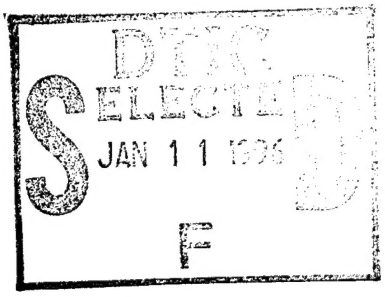
PL-TR-95-2110

ATMOSPHERIC EMISSION AND ANALYSIS EXCEDE III SPECTRAL PROCESSING

R. E. Murphy

Research Sciences Corp.
594 Marrett Rd., Suite 13
Lexington, MA. 02173



June 1995

Scientific Report No. 1

PHILLIPS LABORATORY
Directorate of Geophysics
AIR FORCE MATERIEL COMMAND
HANSCOM AIR FORCE BASE, MA 01731-3010

This technical report has been reviewed and is approved for publication

FRANCIS X. ROBERT
Contract Manager

STEPHAN D. PRICE
Branch Chief

ROGER VAN TASSEL
Division Director

This report has been reviewed by the ESC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/TSI, Hanscom AFB, MA 01731-3010. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JUNE 1995	Scientific No. 1

4. TITLE AND SUBTITLE
ATMOSPHERIC EMISSION AND ANALYSIS-
EXCEDE III SPECTRAL PROCESSING

5. FUNDING NUMBERS
Contract No:
F19628-93-C-0047
PE 62601F
PR 3054
TA GG
WU AA

6. AUTHOR(S)
R.E. MURPHY

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
RESEARCH SCIENCES CORP.
594 Marrett Rd., Suite 13
Lexington, MA 02173

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
PHILLIPS LABORATORY
29 Randolph Rd
Hanscom AFB, MA 01731-3010
Contract Manager: Frank Robert/GPOB

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
PL-TR-95-2110

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

EXCEDE III, a rocket-borne experiment, made measurements of electron irradiated air over altitudes of 80 to 115 km. High spectral-resolution, infrared measurement of the excited air were made with a Michelson type interferometer-spectrometer. The recorded data from the interferometer-spectrometer require a significant amount of computer processing to convert the data into a useful form and provide spectral intensity as a function of wavenumber. A software package to perform the required computer functions has been developed which permits direct interactive capability. The software has been made very concise and easily fits on a personal computer. The programs may be used to set filter band edges, to pre-process interferograms, co-add any number of interferograms, perform Fourier transforms to recover spectra, allow for rectangular, triangular or Kaiser-Bessel apodization, interpolate the spectral output and generate multiple formatted output. Examples are provided using the EXCEDE III data which illustrate the implementation of the programs. The computer program package has been installed and demonstrated at GL/GPO.

14. SUBJECT TERMS			15. NUMBER OF PAGES
Infrared	Radiation	Computer Codes	66
Backgrounds	Spectra	Software	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

FOREWORD

The Phillips Laboratory, Geophysics Directorate Environmental Effects Division PL/GPO has conducted a number of field experiments designed to collect information on the infrared characteristics of the earth-atmosphere backgrounds. The successful collection of the infrared background data constitutes the major effort in a field measurement program, however, it must be followed by an intensive activity to reduce the recorded data to a useful form in order to perform analysis and extract the scientific parameters which will characterize the background emissions. This scientific report describes the generation of a computer capability which greatly aids in the processing of data collected under the EXCEDE III program allowing for rapid generation and hands-on interaction with the high resolution spectra.

EXCEDE III was one of several PL/GPO field experiments which were designed to obtain infrared measurements of the atmosphere under the influence of perturbing energetic electrons. The EXCEDE III program utilized a rocket borne electron accelerator to inject a beam of energetic electrons into the atmosphere at altitudes extending over the 80 - 115 km regime. The electron induced radiation was observed by an number of sensors, all of which performed as intended, including a multi-channel, high resolution interferometer-spectrometer. The nature of the data collected by the interferometer-spectrometer is such that the data must be Fourier Transformed to obtain intensity spectra in the frequency (reciprocal wavelength) domain. Several additional computer manipulations of the data are also required to produce the intensity spectra in a useful form. Where the digital data base is very large, as with the EXCEDE III interferometer, it is highly desirable to have direct and interactive capability to perform many operations on the raw data and perform multiple analytical functions. In concert with the pprincipal investigator for EXCEDE III software for a small personal computer has been developed which accomplishes this need.

This Scientific Report presents a brief introduction to the instrumentation flown on the EXCEDE III experiment with emphasis on the interferometer-spectrometer. The digital data obtained from the interferometer is discussed and the magnitude of the data base is presented. The mathematics required to process the interferograms and recover the spectra is reviewed. The computer software developed to provide a self-contained, personal-computer work-station to operate on the EXCEDE III interferograms is described in detail. Examples of the operations are provided. The complete computer code is provided in hard copy and an executable version has been installed at PL/GPO.

ACKNOWLEDGEMENTS

We wish to express our appreciation to the members of the staff at the Phillips Laboratory/Environmental Division for their support and interest in the research conducted under this contract. In particular, we take this opportunity to thank Dr. Duane Paulsen, Mr. Floyd Cook, Mr. Arthur Giannetti and Mr. Frank Robert for their advice and help derived from several discussions on technical issues.

Thanks are due to all of the members of the EXCEDE III experiment team for their dedicated efforts in acquiring the data. A special note of thanks is due Alan Thurgood of Utah State University.

We are indebted to Dr. W.G.D. Frederick of BMDO/SN for his continued guidance, support and enthusiasm for this work.

1. INTRODUCTION

A major field measurement program was sponsored by the Defense Nuclear Agency in conjunction with the Strategic Defense Initiative Organization and carried out by the Phillips Laboratory, Geophysics Directorate, with support from several contractors. This program, EXCEDE III, was a follow-on to several earlier experiments which explored the phenomenology associated with the interaction of electrons with the atomic and molecular species in the earth's atmosphere [1]. The latest experiment in this series, EXCEDE III, was planned to extend the measurement data base using improved sensors and to utilize a more intense and stable means of exciting the earth's atmosphere. The experimental goal was to obtain detailed spectral and temporal measurements of emissions produced as a result of electrons impacting with the earth's atmospheric ambient species at altitudes above 80 km. The dose rate by the electron beam was much higher (by several orders of magnitude) than would be expected for auroral dosing in order to have sufficient signal levels from the relatively small excitation volume. This high dosing rate allowed investigation of excitation processes involving more than one excited state and extended the number of excitation processes that could be studied by the EXCEDE III experiment. The advantage of the EXCEDE III experiment was that the excitation conditions produced by the electron beam could be controlled in current, energy and time. Thus, the intensity of the radiation induced by the electrons could be referenced to the input dose and the radiative response of the atmosphere to perturbation by energetic electrons could be calibrated.

Considerable effort was expended in the development of a stable, high power, electron accelerator which could be carried aloft by a rocket. Similarly the optical sensors, covering particularly the infrared, but also the visible and the ultraviolet, were designed to match the expected EXCEDE III radiances. In addition, diagnostic packages were assembled to provide an accurate determination of the performance of the system and to define the exact conditions of the measurements. Two separate payloads were used so that the primary instruments could be far enough away from the beam to get a good aspect on the excited atmosphere. The field-of-view, FOV of the primary instruments intercepted the beam far enough from the accelerator module so that nearfield effects, collective effects, and contamination by degassants were all

minimal. The successful preparation and launch of the payload was the result of a very large effort on the part of many experts all working toward a common goal.

The remaining task associated with the EXCEDE III program, following its very productive flight, is the reduction, analysis and interpretation of the data obtained. As with the hardware implementation, scientists and engineers must concentrate on each individual instrument to extract all possible information which relates the emissive properties of the perturbed atmosphere relative to the excitation source. This report is concerned with the infrared emission measured by the Michelson interferometer used to obtain high resolution spectra of the excited atmosphere. Because of the unusual nature of this instrument, several computer intensive activities are required to process the data. In fact, the analysis needs have dictated that a special set of computer programs be developed to assist in the retrieval of the intensity spectra from the raw data. To this end interactive programming has been used to process the raw data on a standard personal computer.

This report briefly describes the interferometer spectrometer and electron beam flown on the EXCEDE III experiment. The fundamental physics related to the recovery of spectra from data taken by an interferometer is briefly described. The computer processing of the raw data is presented to establish the requirements which must be satisfied by the software. The details of the software programs are presented and examples of how to use the system are given. The ease of application and the simplicity of the instructions makes the system very useful for the analysis with immediate interactive control. Although the software is specifically orientated to the needs of the EXCEDE III data processing, the programs are sufficiently general so that they can easily be adapted to other experiments employing Michelson-type interferometers.

2. OVERVIEW OF THE EXCEDE III EXPERIMENT

EXCEDE III was a rocket-borne experiment which was intended to obtain information on the radiative response of a small volume of air in the upper atmosphere to dosing by energetic electrons [2]. In general, the intent was to collect data on the radiative processes which are

produced as the high energy electrons impart their energy to the ambient atmospheric species. The principal interest was centered on the infrared emissions from 2.5 to 20 μm and the ultraviolet/visible from 0.13 to 0.8 μm. High resolution spectra of the infrared emission were required to establish the radiating species, intensities, production and loss sources, and altitude dependance. The visible and ultraviolet emissions were measured to assist in establishing the energy deposition, initial energy spatial distribution and to provide a reference for the determination of the energy efficiency for individual infrared radiating species.

The EXCEDE III payload was made up of two modules, the electron accelerator (electron gun) module and a second module which contained the majority of the optical sensors. The payload was separated from the booster rocket and the modules oriented and separated at a velocity of about 3 meters per second. The gun module was pointed so that the injected electron beam was up the earth's magnetic field lines and the sensor module pointed about 50 meters along the beam from the gun early in the experiment. Figure 1 illustrates the payload concept and geometries involved. The accelerator system consisted of two separate accelerator bays, each powered independently. The electron gun could be operated at 2.6, 2.1, and 1.5 kV so that if a restart was required it would do so at a lower voltage. The gun design was intended to produce an uncollimated beam of electrons at approximately 2.6 kV and 18 amps. At this level the electron dosing gave rise to a peak electron density approaching 10^9 per cubic centimeter which is several orders of magnitude larger than is found in an intense aurora. In addition to the electron gun, this module also carried a Circular Variable Filter spectrometer covering 2.5 - 22μm, film cameras, 6 photometers, a retarding potential analyzer and an electrostatic analyzer; all used to quantify the electron beam performance and the beam environment.

The sensor module contained three CVF spectrometers, ultraviolet and visible spectrometers, an oxygen sensor, two video cameras, an x-ray detector, a scanning photometer package (which swept over the beam) and a Michelson interferometer spectrometer. The three CVF spectrometers were oriented so that one pointed at the beam, one pointed 8 degrees into the afterglow and the third 20 degrees into the afterglow. This configuration of CVF's allows the time history of the emissions to be followed much in the same fashion as a laboratory flowing

afterglow experiment. The interferometer was pointed at the beam to observe the prompt radiating species with high spectral resolution.

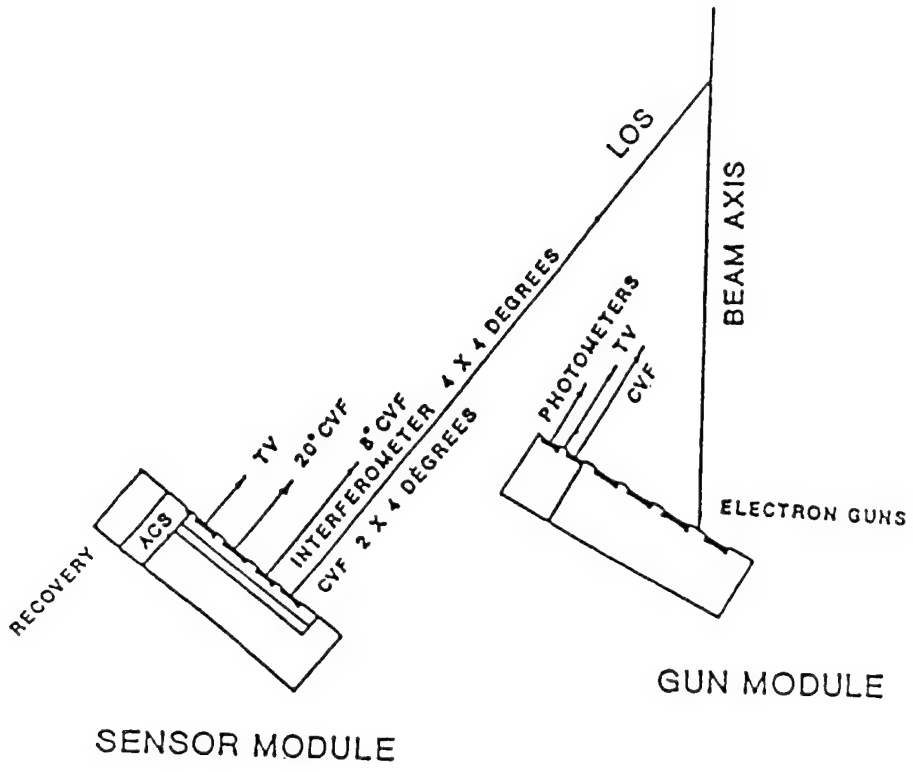


Figure 1. EXCEDE III Payload Showing the Sensor Module and Gun Module.

An Aries rocket carrying the EXCEDE III payload was launched from the White Sands Rocket Range at 01:00:59 MDT, April 27, 1990. The launch was from north to south in order to direct the trajectory parallel to the earth's magnetic field lines at one point on upleg. At this point, (t_0 + 141.7 s) motion of the electron beam across the magnetic field lines is minimized and maximum dosing by the electron beam is achieved. Thereafter velocity of the accelerator and the beam across the field lines increases linearly according to the relation $v_\perp = 4.57(t - 141.7)$ m/s. The payload was separated from the booster at 68 km and followed a ballistic trajectory reaching an apogee of 115 km. Following booster separation the sensor and gun modules were separated from each other at 82 km with a relative velocity of 3.17 m/s. The electron gun and all sensors operated in an outstanding fashion even exceeding expectation.

The first pulse by the electron gun occurred at 120.419 seconds after launch with an active period of 4.730 seconds followed by a beam off period of 2.370 seconds. The first two pulses were unstable but midway into the third pulse the beam became more stable and performed satisfactorily for the rest of the flight. Figure 2 shows the electron beam current relative to time after launch.

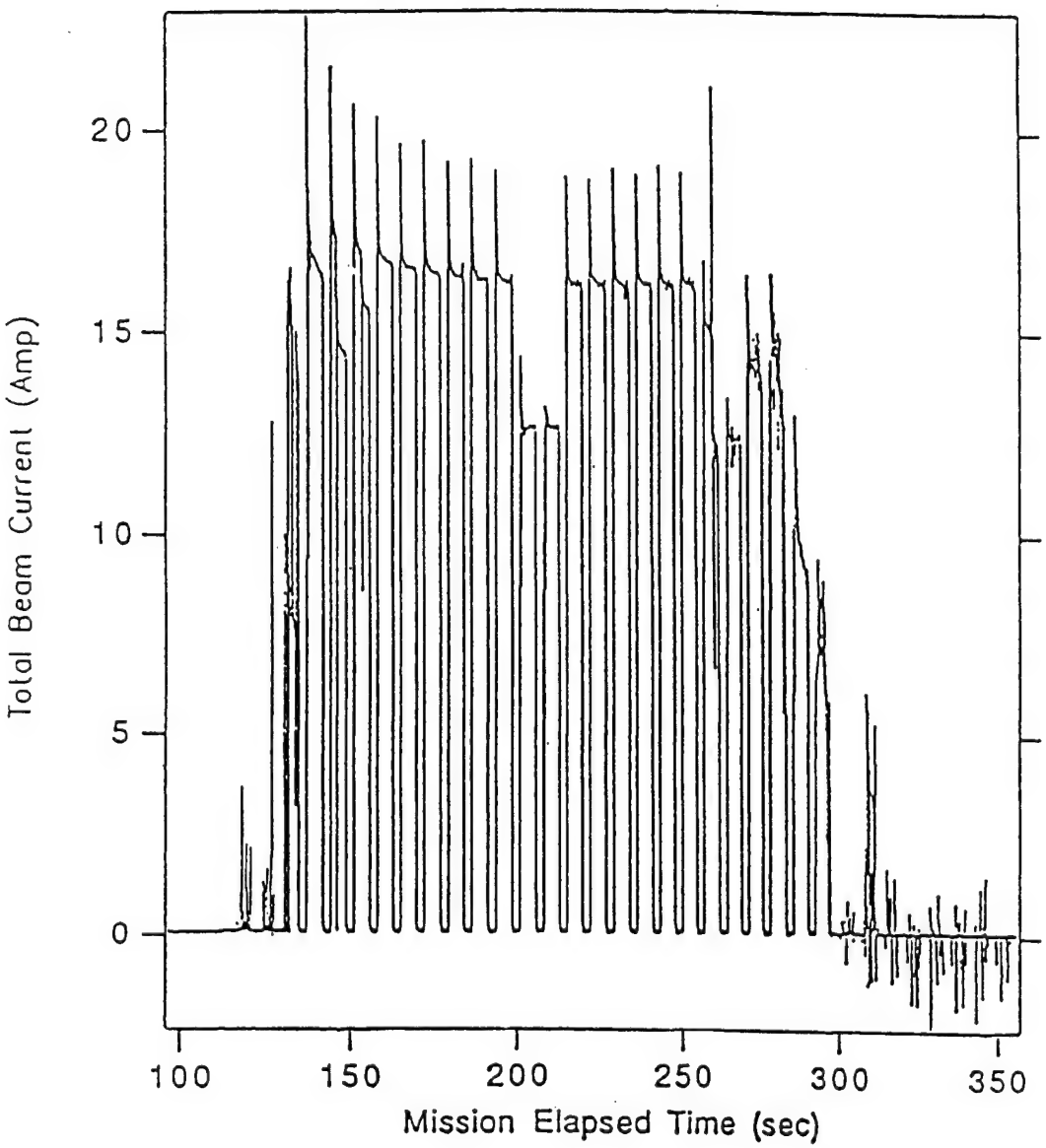


Figure 2. Pulsed Electron Gun Beam Current Versus Mission Elapse Time or Altitude.

The altitude of the payload and the timing for the triggering of the electron gun is respectively given by,

$$h = 115.0 + 0.004732(t - 195.0)^2 \quad \text{km} \tag{1}$$

$$t = 113.319 + 7.100 \times n \quad \text{relative to } t_0 \tag{2}$$

where t_0 is 7:00:59 UT and n is the number of the gun pulse. As will be shown later the timing sequence is very important for the interferometer measurements. The stability and intensity of each beam pulse must be related to the emission spectrum being measured to give optical infrared efficiency determinations.

3. INTERFEROMETER DETAILS

Since this report is concerned with processing software for the data collected by the interferometer spectrometer flown on EXCEDE III, some detail on the make-up of the interferometer and its operation will be reviewed. The Michelson interferometer spectrometer was built by Utah State University [3]. Figure 3 shows the layout of the interferometer. For the EXCEDE III a dual optical cube design was utilized where separate fore optics brought the incident radiation into the individual optical cubes. The interferometer uses a porch swing type of motion to translate the two moving mirrors. Rotation by the torque motor about the pivot produces motion in the two mirrors in opposite directions. Each beamsplitter divides the radiation falling upon it so that half is transmitted through to its moving mirror and half is directed on to its fixed mirror. The radiation reflected from the fixed and moving mirror is recombined at the beamsplitter and directed onto a dichroic filter/beamsplitter (see Figure 4) which spectrally separates the radiation. The two beams emerging from the dichroic beamsplitter are directed to separate detectors and the signals (interferograms) recorded. Thus, the interferometer contains four channels with four separate detectors. The four spectral channels are spectrally isolated to the regions of 2.2 - 4.0, 4.0 - 7.5, 6.5 - 13.0, and 13.0 - 22.0 μm respectively and designated as channels 1 - 4.

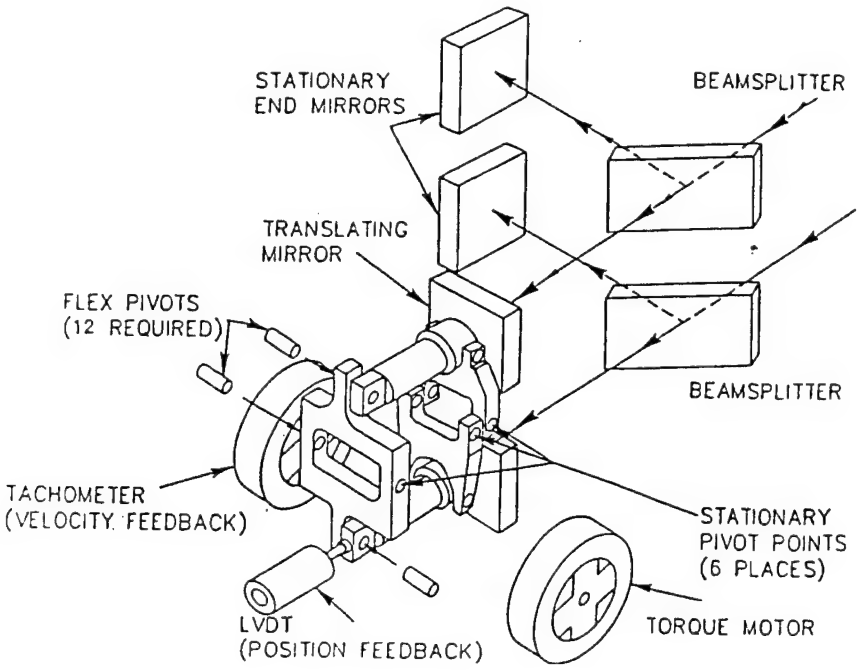


Figure 3. Schematic of the Interferometer Optics and Drive Mechanism.

The spectral channels with specific cut-on and cut-off wavelengths were chosen so that within any one channel only one major atmospheric radiator would be present and, in addition, cover less than one octave. This selection of band limits aids in the dynamic range requirements in a single channel and assists in suppressing nonlinear effects present in detectors and their electronics.

Figure 4 shows more detail of the optical paths in the interferometer and the connection of the moving mirrors. The optical distance traveled by the two beams differs by twice the differential distance between the beamsplitter and the respective mirrors. When the respective mirrors are both equidistant all the recombined radiation is in phase; this is referred to as zero optical path difference, ZOPD. As the optical path difference is varied by the movable mirror different frequencies of the radiation go in and out of phase giving rise to an interferogram. Essentially, an interferogram is the autocorrelation function of the incident radiation falling on the aperture of the interferometer.

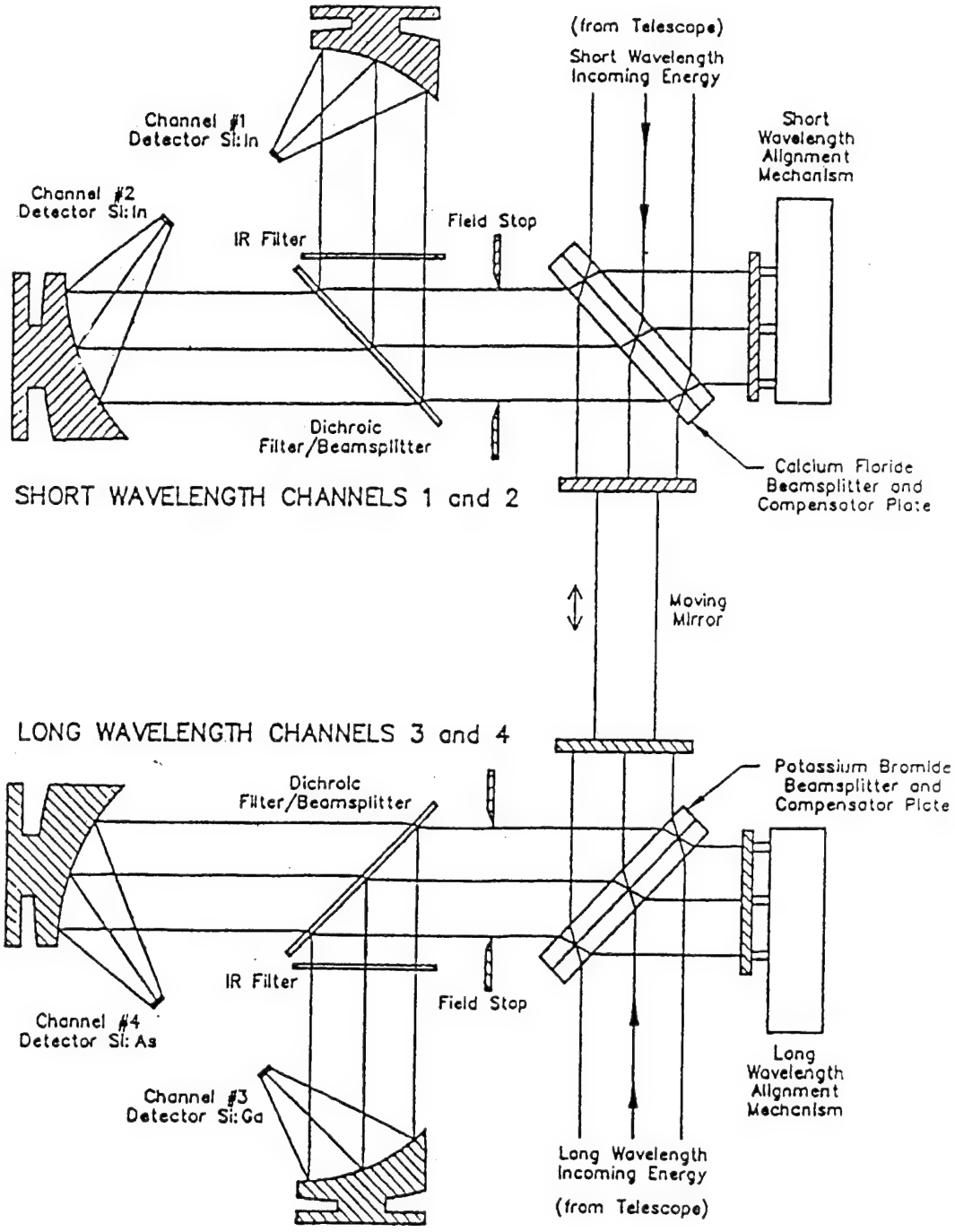


Figure 4. EXCEDE III Dual Optical Cube Interferometer Schematic Layout.

An interferogram obtained from an interferometer contains the multiplexed cosine encoded radiation of the source. It is highly desired that the source be constant over the time duration required to obtain the interferogram. Since the EXCEDE III electron gun was pulsed it was necessary to synchronize the mirror motion to the timing of the pulses. When the electron beam was initiated, the moving mirrors completed two full length scans. During the electron beam off condition the moving mirrors completed one full length scan and a short retrace so as to be in position for the next beam on pulse. Figure 5 gives the approximate timing sequence for the electron gun and a movable mirror.

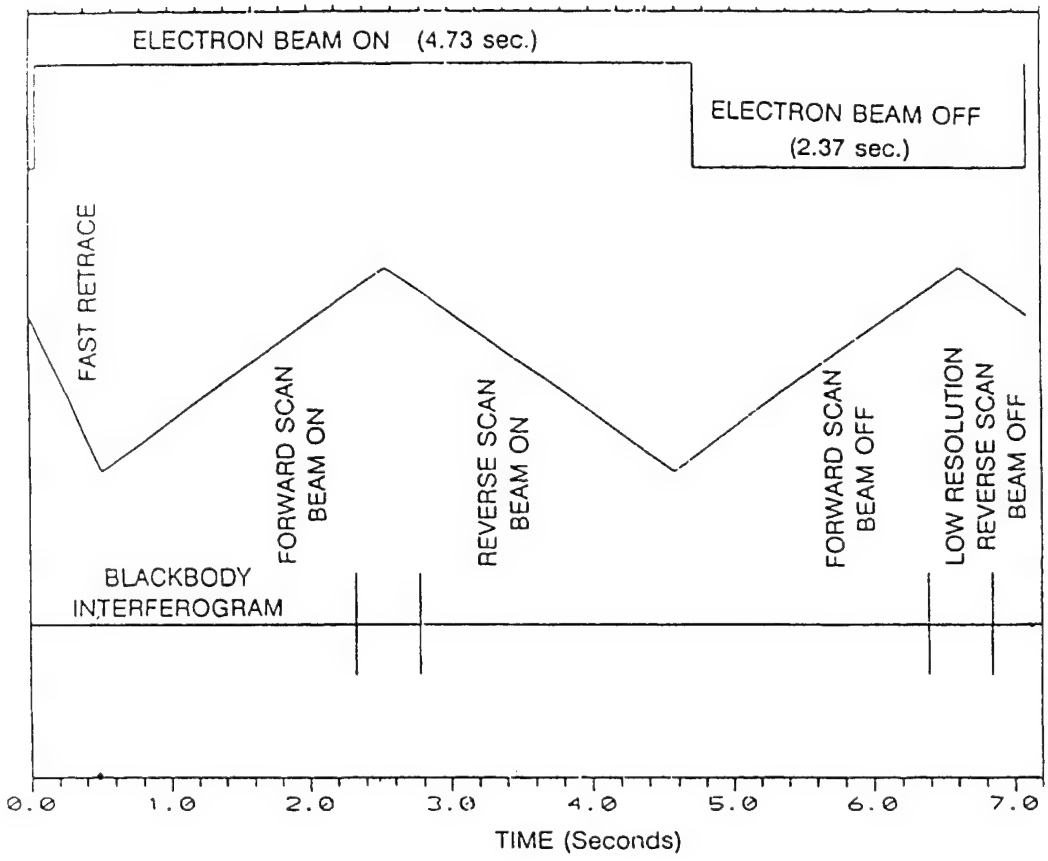


Figure 5. Timing of the Interferometer Scans Synchronized to the Electron Beam.

The moving mirrors were driven so that single sided interferograms were obtained. Ideally, an interferogram is symmetric about the ZOPD. Practical considerations forced the EXCEDE III system engineer to design for single sided interferograms by eliminating almost all of one side of the scan. Enough of the second side of the interferogram was obtained so that phase correction could be done and a full, symmetric interferogram could be created. The interferometer detectors produced an analog output interferogram as the mirrors are driven. The analog interferograms (channels 1 and 2) were converted to a digital data stream by digitally sampling the interferogram at the nulls of a reference channel of He-Ne laser radiation (15,798.0117 cm^{-1}). For channels 3 and 4 the sampling distance was twice that of channels 1 and 2 or half the sampling rate of channels 1 or 2. The interferogram data in all channels were over sampled to satisfy the Nyquest sampling theorem and eliminate aliasing. The salient specifications of the interferometer are included in Table 1.

Table 1. EXCEDE III Interferometer Spectrometer Specifications.

General Description:	Michelson interferometer with dual optical cubes and a common flex pivot drive system.
SPECTRAL RANGE:	2.2 um to 7.5 um and 7.0 um to 22 um.
SPECTRAL RESOLUTION:	2 cm-1 apodized
FIELD OF VIEW:	4 degree full angle
SCAN RATE:	2 seconds/scan
SCAN MODE:	Single sided, bidirectional with fast retrace every other scan to synchronize with the electron gun pulse
OFF AXIS REJECTION:	Shaded primary mirror in telescope privides 7 orders of rejection
DETECTOR MATERIAL	Channels 1 & 2 -- Si:IN; Channel 3 -- Si:Ga; Channel 4 -- Si:As
CRYOGENIC DESIGN:	Liquid helium
DYNAMIC RANGE:	16 bit (65535) each data channel
DATA SAMPLING RATE:	4.74 KHz (from laser reference)
SENSITIVITY:	CH #1 2e-11 Watts/cm-2/Sr/cm-1 CH #2 5e-12 Watts/cm-2/Sr/cm-1 CH #3 6e-13 Watts/cm-2/Sr/cm-1 CH #4 3e-13 Watts/cm-2/Sr/cm-1

4. INTERFEROMETER FIELD DATA

The interferogram signals from the interferometer detector electronics were passed through an on-board premodulation filter and then brought down over a telemetry link at 750 kbit per second in NRZ format. The telemetry data was recorded at several receiving sites. The data tapes recorded at the receiving stations were shipped to the Phillips Laboratory at Hanscom AFB, MA. At the laboratory the field tapes were decommutated and computer compatible digital tapes generated. During the course of the flight approximately 100 interferograms were collected from each channel. Channels 1 and 2 produced interferograms with about 9500 data points and channels 3 and 4 gave interferograms with about 5000 data points. The interferograms from channels 1 and 2 contained about 600 points beyond ZOPD and channels 3 and 4 contained about half that number. The interferograms are identified by channel and IRIG time at the beginning of each interferogram which in turn can be related to the electron gun performance and the altitude in the flight.

The raw interferograms on the digitized computer tapes contained relatively few interferograms which had not been contaminated by several unwanted sources. Figures 6a-d show rare examples of the interferograms from channels 1-4 which are free of signal contamination. The majority of the interferograms suffered from defects which could produce serious distortions in the recovered spectra. Figure 7a-d illustrates some of the defects which reside in the interferograms. In Figure 7, identifying letters correspond to various problems associated with an interferogram i.e.,

T trend in the DC values
Z DC requires subtraction ie., zero mean
P effect of an external pulse (gun related)
S spike often introduced by telemetry
R periodic disturbance (also on other instruments)
O offset followed by total drop-out

Some of the difficulties are easily corrected. Others where the disturbance is within the frequency band of the wanted signal require very special attention to restore. In many cases the

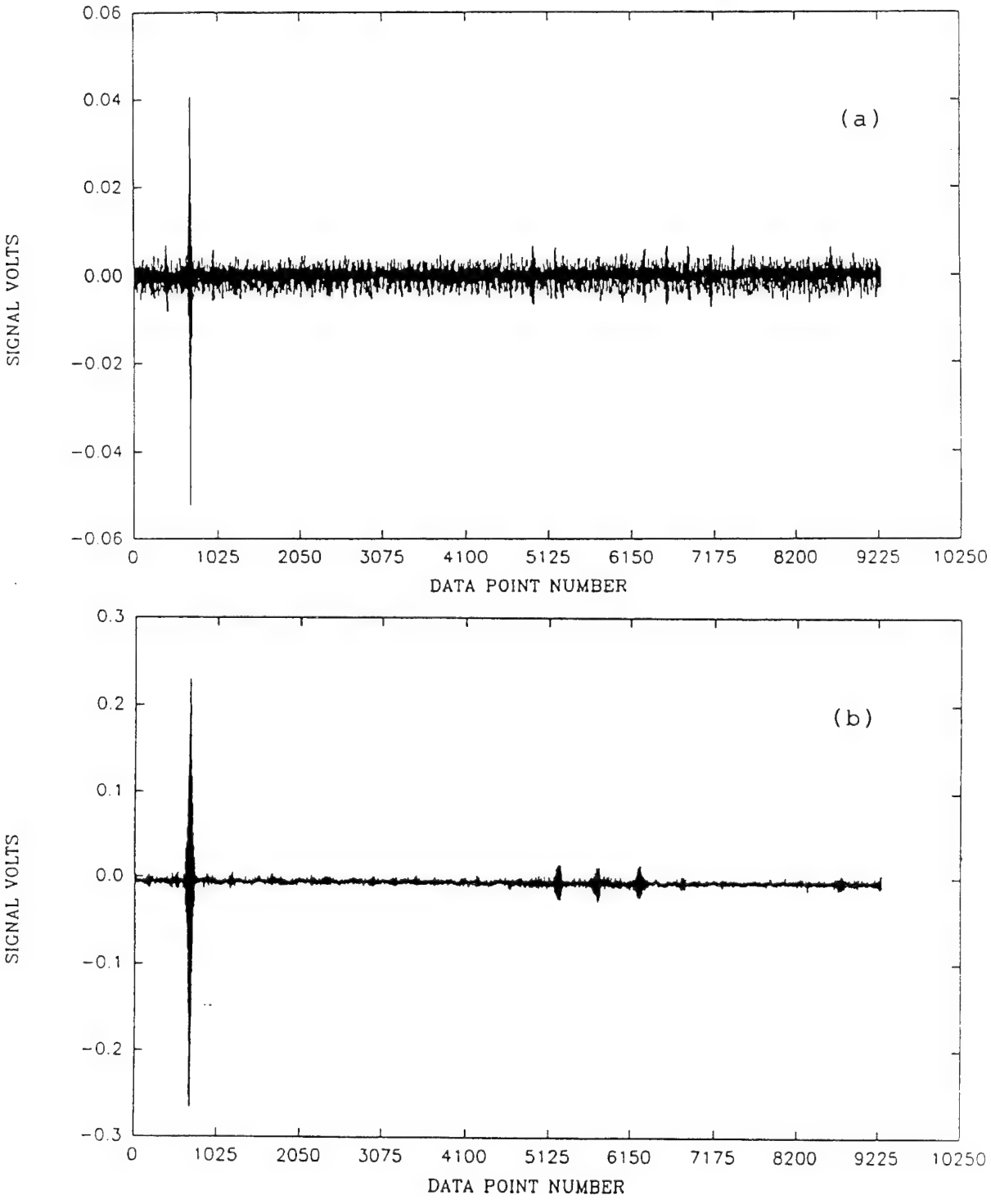


Figure 6 (a,b). Satisfactory Interferograms from Channels 1 and 2 Respectively.

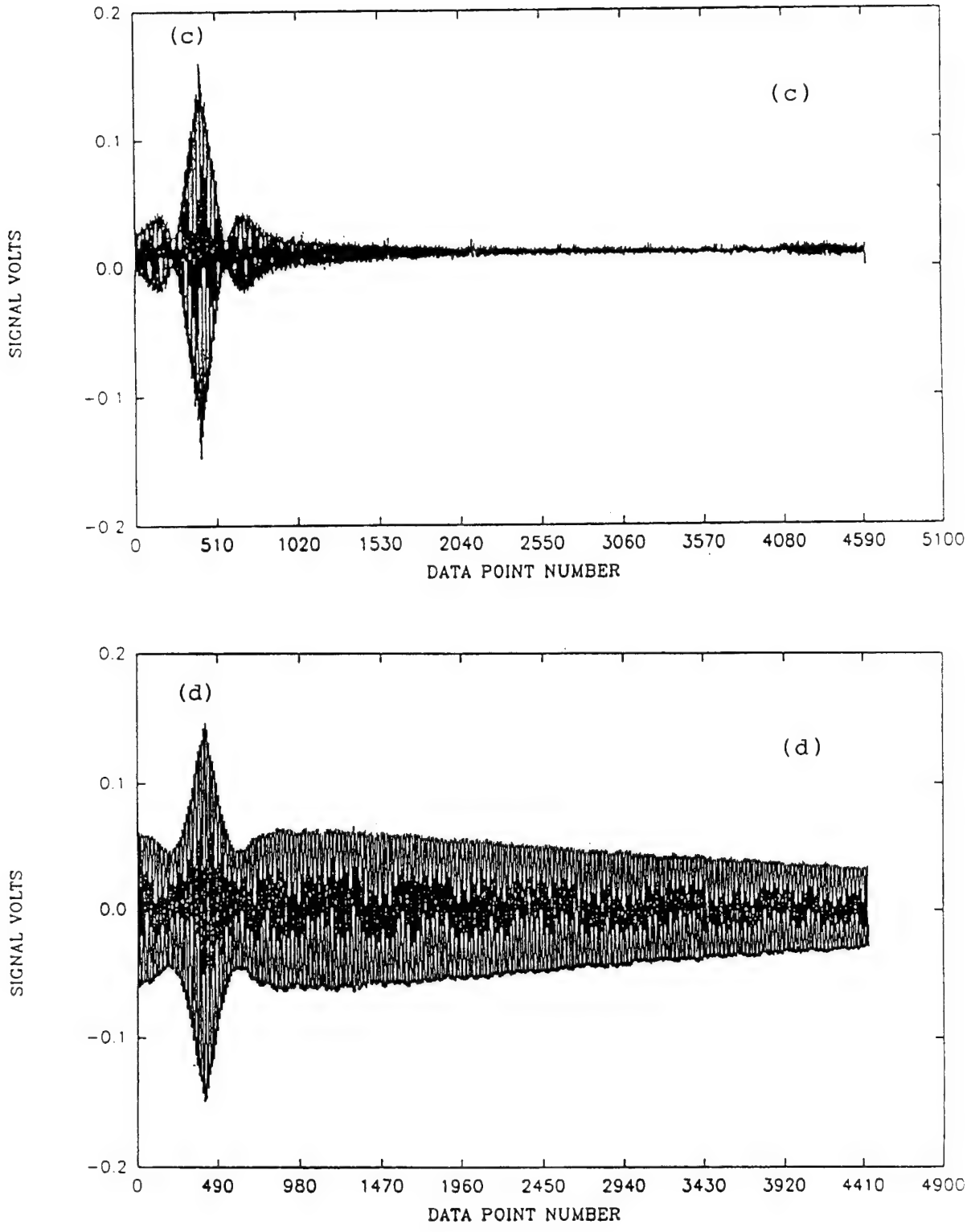


Figure 6 c,d. Satisfactory Interferograms from Channels 3 and 4 respectively.

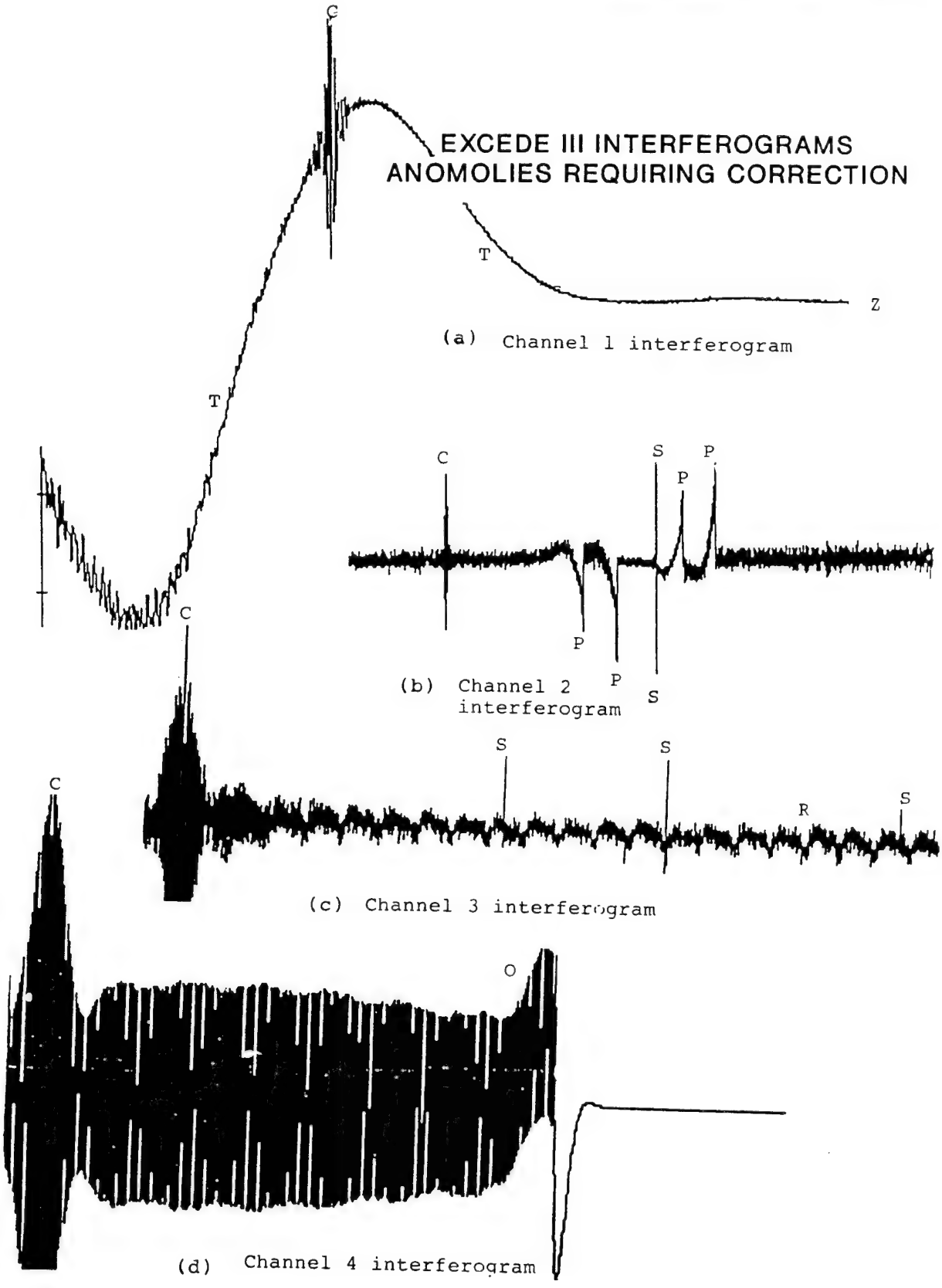


Figure 7 a-d. Interferograms Showing Several Defects Occurring in Channels 1-4 Respectively.

interferograms were so distorted that the ZOPD was nearly indistinguishable, which made corrections even more difficult. Through the use of a number of labor intensive techniques requiring individual treatment of each interferogram, the interferogram database for the EXCEDE III experiment was generated. This database resides at the Phillips Laboratory and is available to interested investigators. However, they should be aware of the problems encountered with the raw data and the operations performed to bring it to its present form. This data base is the starting place to recover the spectra which can be obtained from the EXCEDE III interferograms. Within the data base the interferogram files are designated by channel and IRIG time, for example the interferograms of channel 1 are designated as 10504D.ASC, 10507D.ASC, 10509D.ASC, etc. indicating laser reference channel 1 beginning at IRIG time 7:05:04, Derived from the computer digital tape (including the special manipulations performed to eliminate the problems encountered with the raw data). The interferograms of channel 2 are similarly designated as 20504D.ASC, 20507D.ASC, and so on.

5. SPECTRAL RELATIONS AND THE INTERFEROGRAM

Monochromatic electromagnetic radiation consists of oscillating waves at a specific frequency f (or wavenumber σ, which is equal to $1/\lambda$ with the wavelength λ in centimeters). Consider collimated, quasi-monochromatic radiation of spectral distribution $B(\sigma)d\sigma$ which is divided in half by the beam- splitter of an interferometer (as in Figure 2) and fills the two optical arms (one arm being from the beamsplitter to the fixed mirror and the other arm from the beamsplitter to the movable mirror). If the interferometer arms are of equal length, say $m\lambda$ where m is any integer, then an oscillating wave would 'fit' in both arms identically and upon recombination after being reflected from the mirrors would be reinforced or be in phase. This is the condition of ZOPD. If on the other hand one arm was shorter (or longer) by $\lambda/4$ the radiation in that arm, after reflection, would have traveled less (or more) by the distance $\lambda/2$ upon returning to the beamsplitter. For this condition waves from the two arms recombining at the beamsplitter are completely out of phase and nullify one another. For distances in between these two conditions the amplitude of the recombined waves take on intermediate values depending on their relative phase or alternatively the path difference x in the two arms. As the movable mirror is driven

the detector viewing the recombined waves responds to an intensity which goes through a maximum, decreases, reaches a null, and then increases in a sinusoidal manner. The representation of the signal can be written

$$D(x) = B(\sigma)[1 + \cos(2\pi\sigma x)]d\sigma. \tag{3}$$

The detector output or interferogram (after removing the DC level) is a cosine function. If the radiation is made up of more than a single frequency the signal is given by the integral of Eq. (3) over the range of frequencies i.e.,

$$D(x) = \int_0^{\sigma_{max}} B(\sigma)\,[1 + \cos 2\pi\sigma x]\,d\sigma \tag{4}$$

Removal of the DC term from Eq. (4) gives the interferogram F(x),

$$F(x) = \int_0^{\sigma_{max}} B(\sigma)\cos 2\pi\sigma x d\sigma = Re\int_0^{\infty} B(\sigma)\, e^{i2\pi\sigma x} d\sigma \tag{5}$$

If $B(\sigma)$ were an even function the spectral density distribution could be obtained by a cosine transformation.

$$B(\sigma) = \int_0^{\infty} F(x)\cos 2\pi\sigma x\, dx \tag{6}$$

Generally, $B(\sigma)$ is not an even function so for mathematical convenience, negative frequencies are introduced although they have no corresponding physical reality. Any function can be written as the sum of its even and odd parts i.e., $B(\sigma)$ is the sum $B_e(\sigma) + B_o(\sigma)$. Further, the even and odd functions may be expressed as

$$B_e(\sigma) = \frac{1}{2}[B(\sigma) + B(-\sigma)], \quad B_o(\sigma) = \frac{1}{2}[B(\sigma) - B(-\sigma)] \tag{7}$$

$$F(x) = \int_{-\infty}^{\infty} B_e(\sigma)\, e^{i2\pi\sigma x} d\sigma + \int_{-\infty}^{\infty} B_o e^{i2\pi\sigma x} d\sigma \tag{8}$$

It is easily seen that the real interferogram becomes

$$F(x) = \int_{-\infty}^{\infty} B_e(\sigma)\, e^{i2\pi\sigma x} d\sigma = \int_{-\infty}^{\infty} B_e(\sigma)\, cos2\pi\sigma x d\sigma \tag{9}$$

In practice the interferogram is truncated by the limits of the movable mirror, over the maximum distance X. The truncation function T(x) is unity between -X and X and is zero elsewhere and the finite interferogram is

$$F(x) = \int_{-\infty}^{\infty} T(x)\, B_e(\sigma)\, e^{i2\pi\sigma x} d\sigma \tag{10}$$

and

$$B(\sigma) = \int_{-\infty}^{\infty} T(\sigma')\, F(\sigma-\sigma')\, d\sigma' = \tilde{T}(\sigma) * \tilde{F}(\sigma) \tag{11}$$

where * denotes convolution and $T(\sigma)$ is $2X\text{sinc}(2\sigma X)$ and the ~ indicates the Fourier transform.

For an ideal interferometer no phase distortion of the interferogram occurs and it is symmetric so that a cosine transform will recover the spectral density distribution. However, in most cases the interferometer optical components introduce a phase shift which is nonlinear

and frequency dependent. If the interferometer movable mirror is driven equi-distant from the ZOPD, from -X to X, the spectral density is the modulus of absolute value i.e.,

$$B(\sigma) = (C^2 + S^2)^{1/2} \tag{12}$$

where C and S are the cosine and sin transformations respectively.

In the EXCEDE III interferograms there was considerable phase distortion. The EXCEDE III interferometer was operated in a single sided mode where the optical path difference was scanned from the maximum optical path difference X through ZOPD to a small distance beyond ZOPD. For single-sided interferograms, the interferogram is represented by

$$F(x) = \int_{-\infty}^{\infty} B(\sigma)\, e^{i[2\pi\sigma x + p(\sigma)]}\, d\sigma \tag{13}$$

with the values of F(x) confined within -dX to X. The phase function can be obtained from a short double-sided interferogram. The Fourier transform of the interferogram extending from -dX to dX is

$$\int_{-\infty} B(\sigma)\, e^{ip(\sigma)}\, sinc[2dX(\sigma' - \sigma)]\, d\sigma \tag{14}$$

where $B(\sigma)$ is an even function with the asymmetric nature being explicitly related to the phase function produced by the optical components of the interferometer. Due to the physical nature of the optical components, the expression $e^{ip(\sigma)}$ varies very slowly compared to sinc $(2X\sigma)$ and may be taken outside the integral,

$$e^{ip(\sigma)} \int_{-\infty}^{\infty} B(\sigma)\, sinc[\sigma X(\sigma' - \sigma)]\, d\sigma \tag{15}$$

The phase from a double-sided interferogram is simply the arctan of the imaginary part divided by the real part or

$$p(\sigma) = \frac{\int F(x)\sin(2\pi\sigma x)\,d\sigma}{\int F(x)\cos(2\pi\sigma x)\,d\sigma} = \arctan(S/C) \tag{16}$$

The phase function may be obtained from a single-sided interferogram by performing the sine and cosine transform over the very short optical path difference -dx to dx. Note that $B(\sigma)$ is the Fourier transform of the measured interferogram modified by the apodization function

$$B(\sigma) = Real\ e^{[-ip(\sigma)]}\ B_u(\sigma) \tag{17}$$

An equivalent expression for $B(\sigma)$ may be written as,

$$B(\sigma) = |B(\sigma)|e^{i\beta(\sigma)} \tag{18}$$

and the expression for the phase-corrected spectrum becomes,

$$B_c(\sigma) = |B_u(\sigma)|\ Re\ e^{[i\beta(\sigma) - p(s)]} \tag{19}$$

or

$$B_c(\sigma) = |B_u(\sigma)|\cos[\beta(\sigma) - p(s)] \tag{20}$$

which is the more familiar as the Mertz form [6].

A ramp function is used to calculate the phase corrected spectrum which consists of a linear, left ramp of short extent which passes through the chosen ZOPD with a magnitude of one half. The ramp extends equally on either side of the ZOPD with magnitude from zero to unity and then extends with a constant value of one to cover the full length of the interferogram. The ramp portion essentially multiplies the interferogram on either side of the ZOPD so as to eliminate counting the information about ZOPD twice. An apodizing function which has a value of one-half everywhere along the ramp function and abruptly becomes unity where the ramp first takes the value of one would also account for proper counting of information. However, the abrupt nature of this step function introduces unwanted effects in the instrument line shape and is therefore not acceptable. The ramp function described earlier is used in the programs which follow.

It should be noted that a more general phase function, $\phi(\sigma)$ may contain both phase slip due to improper selection of the ZOPD in addition to the phase introduced due to the optical components of the interferometer. The phase for these two effects is given by

$$\phi(\sigma) = p(\sigma) + 2\pi\epsilon\sigma \tag{21}$$

where ϵ is the displacement from the true ZOPD. The choice of an offset ZOPD is accommodated in the process of determining the phase function from the short, double-sided interferogram. If the chosen ZOPD is selected with a large displacement relative to the true ZOPD, distortion to the recovered spectrum can occur. For the EXCEDE III interferometer the mechanics producing the mirror turn around points were very reproducible and the ZOPD occurred within a few points of the same position for forward scans and similarly, but with a different position, for the reverse scans. Consequently, selection of the ZOPD was never very far from the true position and spectrum errors from this problem are easily avoided.

6. DESCRIPTION AND USE OF THE SOFTWARE

Up to this point, it has been shown that the EXCEDE III data obtained from the

interferometer spectrometer contained many faults of various origins which were recorded on the field data tapes. These anomalies were preserved in converting the field tapes to computer compatible digital tapes. With considerable individual attention given to each interferogram a corrected version of each interferogram was generated. Digital files of all corrected interferograms are kept at PL/GPO. These data files easily fit into a modest personal computer. The material in this section deals with the capabilities which have been developed to investigate the spectral information which is contained in the corrected data files.

The directory of files (Fortran progrms are given in Appendix A) for the software utility developed for use with the EXCEDE III interferometer data is :

```
   Directory of C:\REM

.                <DIR>
..               <DIR>
FILTER   DAT       43
CAL1     ASC     24437
CAL2     ASC     15994
CAL3     ASC      9000
CAL4     ASC      4715
DATFILE  DAT       27
COADD    EXE    228050
RANDY    EXE    405230
```

The right-hand column gives the bit count for the respective files. It is seen that the memory requirement for these programs is relatively small and can easily be accommodated by a personal computer. The function of each file is described briefly in the following:.

FILTER.DAT: This is an ASCI file used to set the cut-on and cut-off frequency (in wavenumbers) for each of the four spectral channels of the interferometer. The digital filter permits frequencies to be chosen to isolate a spectral region or may be used to help the signal to noise ratio by eliminating the noise outside the spectral region of interest. This file is shown in Table 2. These values for the filter band edges have been included with the delivery of the software to PL/GPO.

TABLE 2. Filter File

2650	4650
1330	2700
800	1430
465	830

The wavenumber values in the first row corresponds to the long wavelength upper band limit in column one and to the short wavelength band limit in column two for channel 1. Similarly, the rows two through four are the band limits for channels 2 through 4. The numerical values for this file can be modified with any standard editing routine.

CAL1.ASC: This is a calibration file for channel 1. The numerical values in the file are the factors required to convert the output voltages to engineering units. Interferograms are recorded in volts which may be Fourier transformed and then converted with the calibration factors to give the intensity in terms of the engineering units i.e., watts/cm^2 sr cm^{-1}. The CAL files supplied with this software are based on the laboratory calibrations performed by Utah State University (USU) prior to the actual flight of EXCEDE III. It is known that the pre-flight calibrations are slightly inaccurate, however, they are currently the only calibration values available. It is anticipated that a better calibration will become available in the near future. Part of the advantage of this software package is that when a better calibration becomes available, it can be substituted directly into the Cal files. Figure 11 shows the calibration curve generated by USU for channel 1 which has been taken to produce CAL1.ASC. Similarly, CAL2.ASC, CAL3.ASC and CAL4.ASC are files for the conversion to engineering units for channels 2, 3 and 4. Figures 12 through 14 show the USU calibration curves for channels 2 through 4 respectively and have been delivered with the software package.

COADD.EXE: This is a routine which will add the interferogram values at a specific sampling point (optical path difference) for all sampling positions over the complete interferogram to the corresponding values of other selected interferograms. In this fashion, the numerical value in the coadded interferogram is the sum of the individual contributions at each

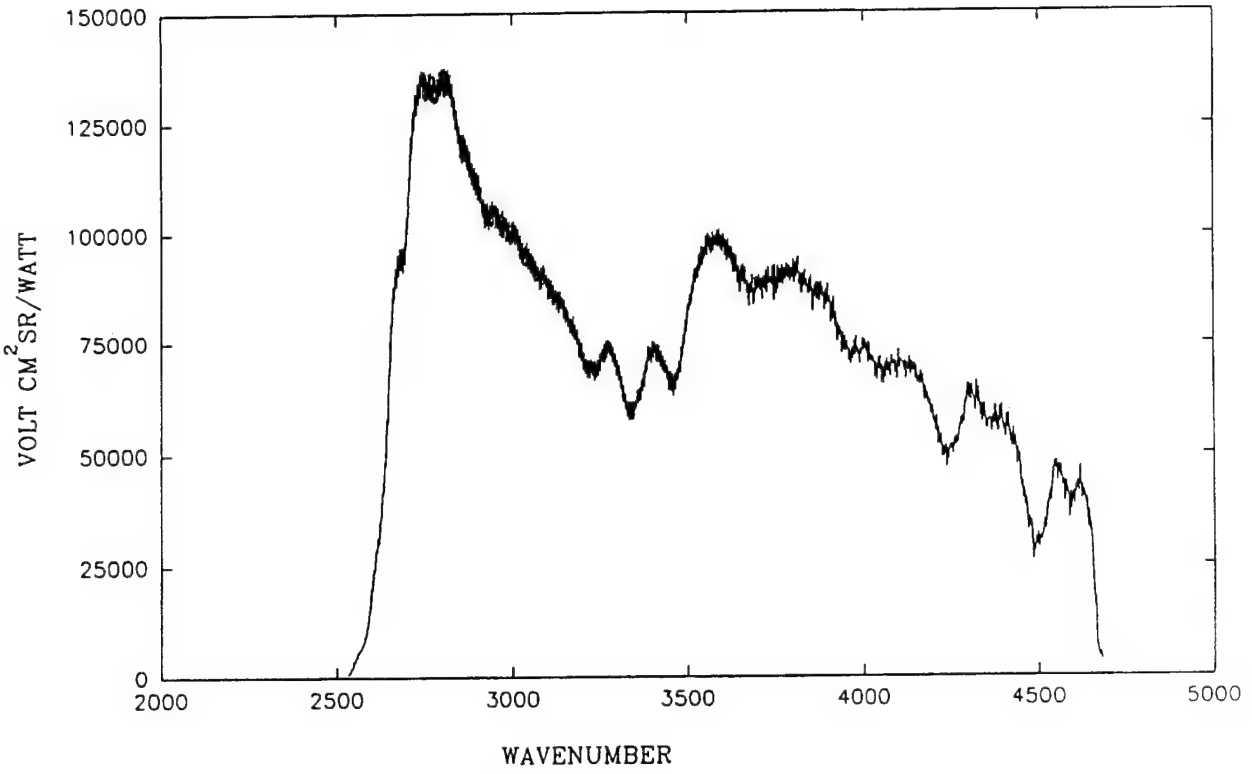


Figure 8. Spectral Response of Channel 1.

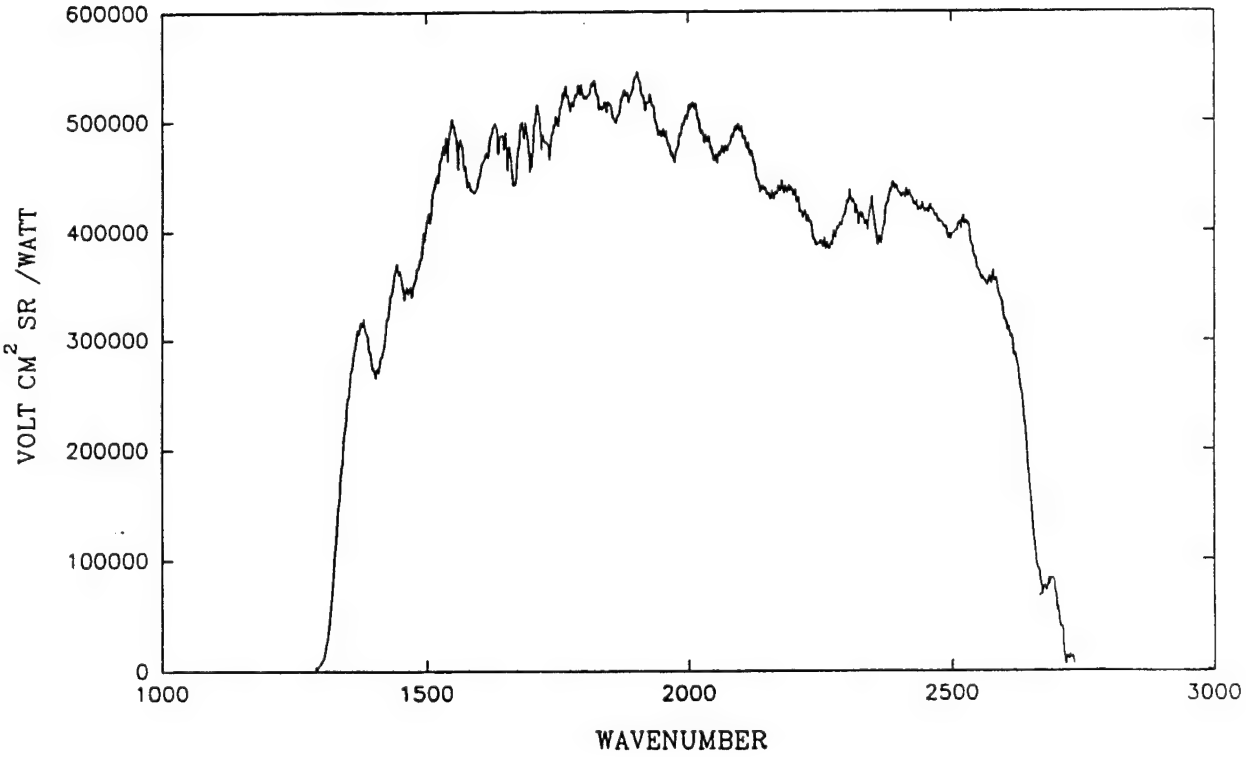


Figure 9. Spectral Response of Channel 2.

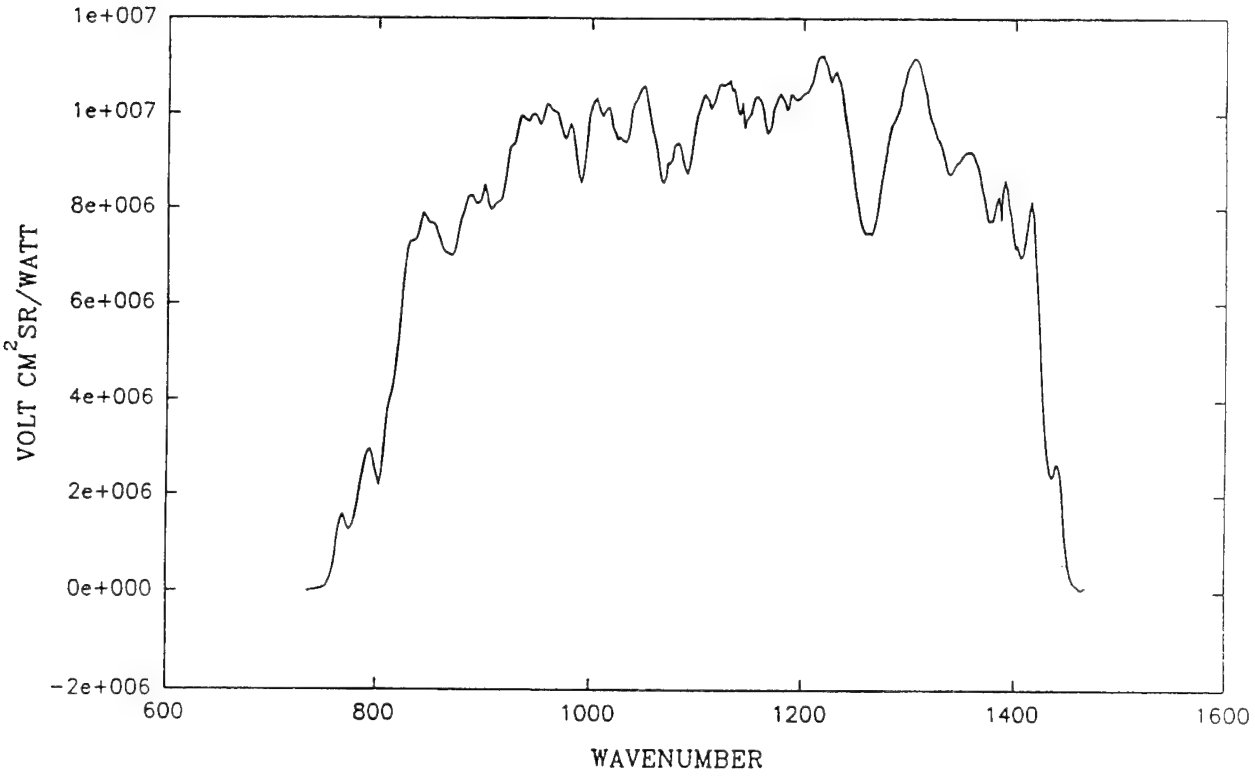


Figure 10. Spectral Response of Channel 3.

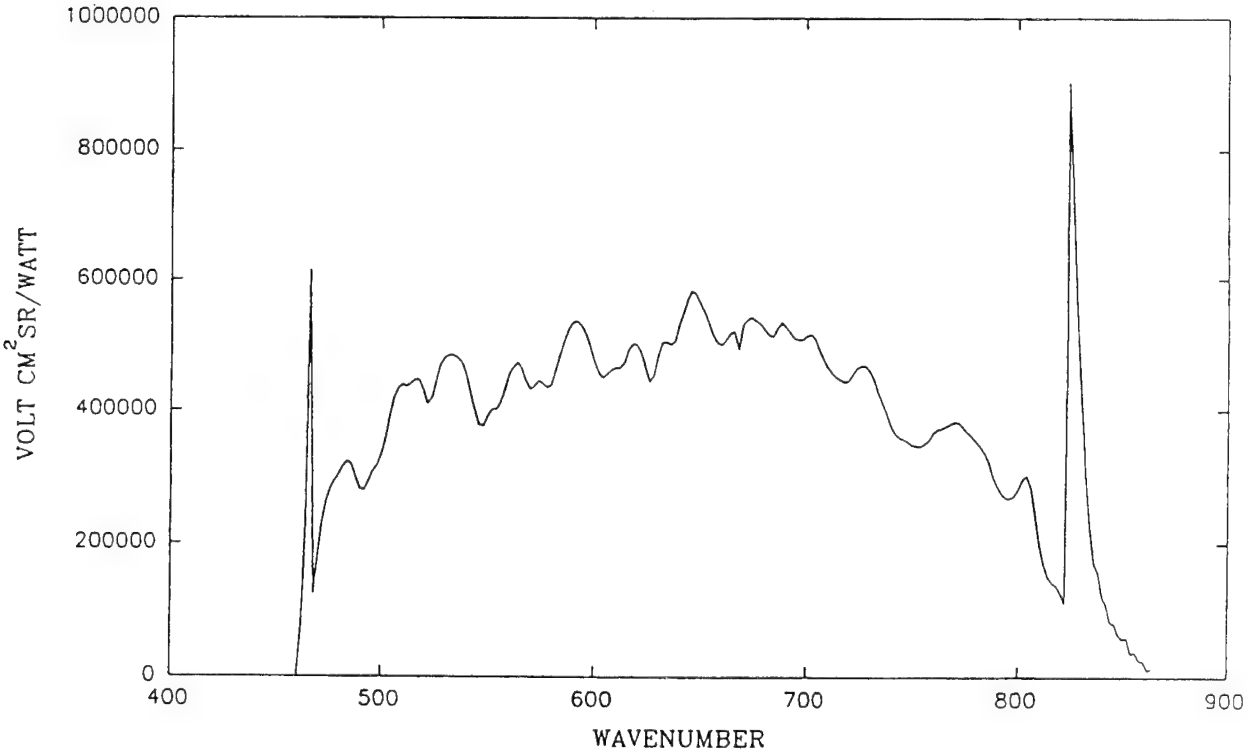


Figure 11. Spectral Response of Channel 4.

sample point for all the interferograms added. Obviously, the signal-to-noise ratio is improved through co-addition. It has been reported that there is a slight advantage in co-adding in the interferogram domain over co-addition in the spectral domain, so the former technique has been adopted in the softwares developed here. The Fortran code for co-adding interferograms is given in Appendix B.

RANDY.EXE: This is an executable file which performs the Fourier transformation of an interferogram file. The routine provides the user with the capability to perform 1) phase correction, 2) Fourier transform an interferogram, 3) get an uncalibrated or calibrated spectrum, 4) select an apodization function including Kaiser Bessel apodization with four values for alpha, and 5) select the number of points to be interpolated between transform output points. The spectral intensity files are produced as an ordered single column of intensity or two ordered columns of wavenumber and intensity.

SAMPLE CO-ADDITION AND PHASE CORRECTION

By way of an illustration the procedure to perform co-addition will be demonstrated with the users input underlined and the computer screen response given. The illustration is for the co-addition of two interferograms but the extension to larger numbers of interferograms is obvious.

Before beginning the program, first select from the archived interferogram data files stored at GP/GPO the interferograms to be co-added. For example, select the interferograms; designated by the asterix i.e., files 20514D.ASC and 20516D.ASC.

```
20511D    ASC    204600
20513D    ASC     42680
20514D    ASC*   199100
20516D    ASC*   204600
20518D    ASC    204600
20520D    ASC     42680
20521D    ASC    204600
20524D    ASC    204600
20526D    ASC    204600
```

Load the selected interferogram files into the Directory REM.

Begin the co-add processes. Comments are denoted as {}

TYPE CD REM(CR) {(CR) denotes carriage return}

REM> RANDY (CR)

Input interferogram file name to read

20514D.ASC

Input channel # (1 - 4)

2

reading 20514D.ASC

9040 points read {gives total no. of points in interferogram}

Interferogram file is 20514d.ing {gives name to interferogram}

Input selection by number

```
*****************************
* 1 : phase correct         *
* 2 : get uncalibrated spectrum *
* 3: get calibrated spectrum *
* 4 : get new interferogram *
* 5 : quit                  *
*****************************
```

1

Phase corrected interferogram is 20514D.cor {Phase corrected file, indexed}

Phase corrected interferogram (1 col) is 20514D.cr1 {Phase corrected single column}

Input selection by number

```
*****************************
* 1 : phase correct         *
* 2 : get uncalibrated spectrum *
* 3:  get calibrated spectrum *
* 4 : get new interferogram *
* 5 : quit                  *
*****************************
```

4 {select the second interferogram to be co-added}

Input interferogram file name to read
20516D.ASC

Input channel # (1 - 4)
2

reading 20516D.ASC
9290 points read

Interferogram file is 20516D.ing

Input selection by number
```
*******************************
* 1 : phase correct           *
* 2 : get uncalibrated spectrum *
* 3 : get calibrated spectrum  *
* 4 : get new interferogram    *
* 5 : quit                     *
*******************************
```

1

Phase corrected interferogram is 20516D.cor {name of phase corrected file, indexed}

Phase corrected interferogram (1 col) is 20516D.cr1 {single column of interferogram values}

Input selection by number
```
*******************************
* 1 : phase correct           *
* 2 : get uncalibrated spectrum *
* 3 : get caibrated spectrum   *
* 4 : get new interferogram    *
* 5 : quit                     *
*******************************
```

5

Stop - Program terminated.

NOTE: The Directory of REM should look as indicated below with the phase corrected interferograms denoted by 20514D.cor and 20516D.cor. These are the phase corrected interferograms which will be used as the input files for use with the routine to co-add interferograms. Obviously, the program can be repeated to include as many interferograms for phase correction as is desired. The files ending with .ING, .CR1 are the truncated original interferogram and the phase corrected interferogram respectively and may be of use to some plotting routines. Otherwise these files may be discarded.

```
   Directory of C:\REM

.                <DIR>
..               <DIR>
FILTER   DAT        43
CAL1     ASC     24437
CAL2     ASC     15994
CAL3     ASC      9000
CAL4     ASC      4715
DATFILE  DAT        27
20514D   ASC    199100
20516D   ASC    204600
20514D   ING    271200
20514D   COR    263550
20514D   CR1    158130
20516D   ING    278700
20516D   COR    271050
COADD    EXE    228050
20516D   CR1    162630
RANDY    EXE    405230
```

Figure 12a shows an interferogram, 20514D, taken from the anomalies corrected files and illustrates the distortion of the interferogram due to the optical components of the interferometer. After using the phase correction software program the interferogram is as shown in Figure 12b. If the phase corrected interferogram shows asymmetry after having been processed by the phase correction routine (for very distorted inteferograms) the processes may be repeated to give further correction. There is a small sacrifice to the total number of data points retained by repeated application of the phase correction process.

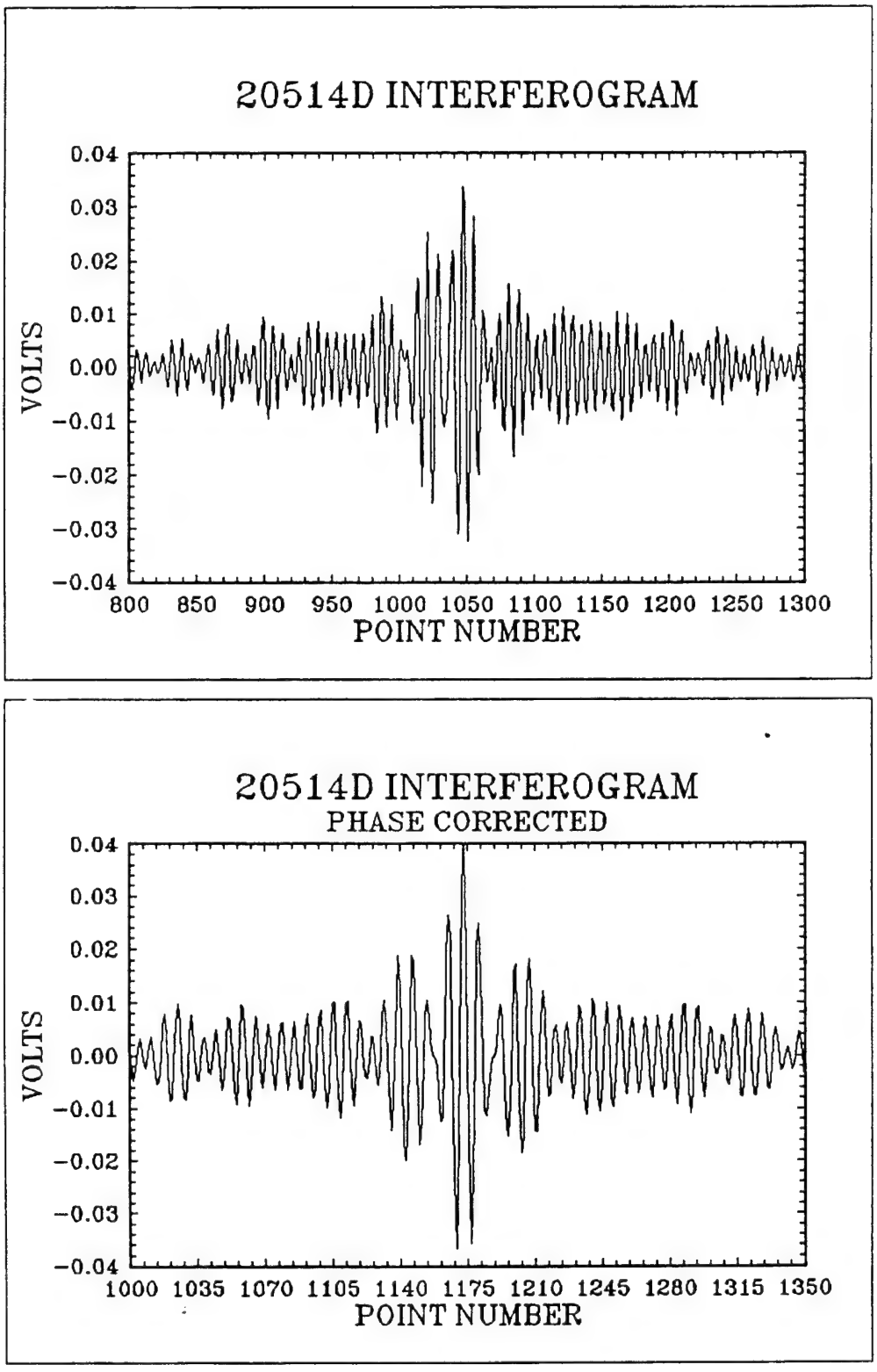


Figure 12. Interferogram a) Before Phase Correction and b) After Phase Correction

At this juncture it is necessary to create a file for the COADD routine which designates for the program which interferograms to co-add. Since we are co-adding the 20514 and 20516 files the required file, "DATFILE.DAT" consists of the simple ASCI file as follows:

```
DATFILE.DAT          27                    {27 Bits}
                2
                20514D.cor
                20516D.cor
```

(if more files were to be co-added the total number would be entered and the file names listed).

DATFILE.DAT must be in directory REM.

With DATFILE.DAT loaded into REM the co-addition routine can be implemented i.e.,

C:> CD REM (CR)

REM>COADD (CR)

```
Reading 20514D.cor
        919          920          7865     {7865 points after the center point at
                                           920 so that there are a total of 8785 points}
        8785 points read
reading 20516D.cor
        9035 points read
         568          569          8466
         568          569          8466          569     {Co-added interferogram
                                 of 8466 points beyond center with center point 569}
```

{When the interferograms are of unequal length the program essentially doubles the values of the longer interferogram where there are no values of the shorter interferogram as in the case above. The average of the value for the co-added interferograms is used.}

```
Writing coadded file OUT1.DAT      {Co-added interferogram as a single column)
Writing coadded file OUT2.ING      {Co-added interferogram as an indexed column}
Stop - Program terminated.
```

The file OUT1.DAT is the co-added interferogram made up from the average of files 20514D and 20516D. The co-added interferogram may be processed just as any other interferogram using the RANDY routines in REM.

NOTE: The Directory of REM should look as indicated in the following:

```
   Directory of C:\REM

.                <DIR>
..               <DIR>
FILTER   DAT        43
CAL1     ASC     24437
CAL2     ASC     15994
CAL3     ASC      9000
CAL4     ASC      4715
DATFILE  DAT        27
20514D   ASC    199100
20516D   ASC    204600
OUT1     DAT    162630
20514D   COR    263550
OUT2     ING    271050
20516D   COR    271050
COADD    EXE    228050
RANDY    EXE    405230
```

8. SAMPLE SPECTRAL RECOVERY

The interferograms to be Fourier transformed must be placed in REM either from the co-add routine just given or by transferring the interferograms from the files at PL/GPO. The spectral recovery process will be illustrated using the co-added interferogram, OUT1.DAT from the previous example.

To execute the spectral recovery routines RANDY is used i.e.,

C:> CD REM(CR)

C:\REM > RANDY(CR)

Input interferogram file name to read {phase corrected, coadded interferogram from the previous example}

out1.dat

```
Input channel # (1 - 4)
2
reading out1.dat
        9025 points read

Interferogram file is out1.ing

    Input selection by number
*******************************
* 1 : phase correct           *        {the uncalibrated spectrum does not apply
* 2 : get uncalibrated spectrum *       the USU engineering conversion,
* 3 : get calibrated spectrum  *        this may be useful when the S/N
* 4 : get new interferogram    *        of the calibration is poor}
* 5 : quit                     *
*******************************

3
Choose apodization by number
*******************************
* 1 : Rectangular             *    {3 types of apodization or spectral
* 2 : Triangular              *        smoothing can be selected}
* 3:  Kaiser Bessel           *
*******************************

1

Input factor to interpolate (1,2,4)    {Output point per input point for the recovered
                                        spectrum can be obtained at  factors of 2 and 4.
                                       The interpolation zero fills the interferogram to
                                      double or quadruple its length.}
2
Array size          65536
Reading cal.file    cal2.asc

Spectrum file is out1.spc       {Spectral output two columns of wavenumber vs intensity}
Spectrum file (1 col) is out1.sc1)  {Spectral output in an ordered single column}
```

Wavenumber file (1 col) is out1.wnc {Wavenumber corresponding to the intensity in out1.sp1]

{These spectral output files reside in REM}

{THIS COMPLETES THE RECOVERY OF THE SPECTRUM.}

{The process may be repeated with different selections for apodization or interpolation. For example, choosing the Kaiser Bessel apodization and interpolation by 4; we have}

```
    Input selection by number
*******************************
* 1 : phase correct           *
* 2 : get uncalibrated spectrum *
* 3 : get calibrated spectrum  *
* 4 : get new interferogram    *
* 5 : quit                     *
*******************************
```

3

```
Choose apodization by number
*******************************
* 1 : Rectangular             *
* 2 : Triangular              *
* 3 : Kaiser Bessel           *
*******************************
```

Input alpha (0 - 3)

3

Input factor to interpolate (1,2,4)

4

Array size 131072

Reading cal.file cal2.asc

Spectrum file is out1.spc

```
Spectrum file (1 col) is out1.sc1
Wavenumber file (1 col) is out1.wnc

    Input selection by number
*****************************
* 1 : phase correct          *
* 2 : get uncalibrated spectrum *
* 3 : get calibrated spectrum *
* 4 : get new interferogram  *
* 5 : quit                   *
*****************************

5

STOP PROGRAM TERMINATED
```

The three output files, i.e., out1.spc, out1.sc1 and out1.wnc are generated to make spectral plotting convenient depending on the input to plotting routines required. For example, out1.spc is easily plotted with the Sigma Plot programs. Out1.sc1 and out1.wnc are more convenient for the DaDISP programs. Both plotting programs are used at PL/GPO. Tables 3-5 show the typical out put from out1.spc, out1.sc1 and out1.wnc.

TABLE 3. Sample of the Output File OUT1.SPC

1330.161000	8.645130E-12
1330.281000	8.203711E-12
1330.402000	7.762762E-12
1330.522000	7.311606E-12
1330.643000	6.842475E-12
1330.763000	6.349588E-12
1330.884000	5.831322E-12
1331.005000	5.288738E-12
1331.125000	4.726105E-12
1331.246000	4.150521E-12
1331.366000	3.573100E-12
1331.487000	3.007161E-12
1331.607000	2.469104E-12

TABLE 4. Sample of the Output File OUT1.SC1

```
8.645130E-12
8.203711E-12
7.762762E-12
7.311606E-12
6.842475E-12
6.349588E-12
5.831322E-12
5.288738E-12
4.726105E-12
4.150521E-12
3.573100E-12
3.007161E-12
2.469104E-12
1.976549E-12
1.549820E-12
1.209811E-12
9.801906E-13
8.784446E-13
```

TABLE 5. Sample of the Output File OUT1.WNC

```
1330.161000
1330.281000
1330.402000
1330.522000
1330.643000
1330.763000
1330.884000
1331.005000
1331.125000
1331.246000
1331.366000
1331.487000
1331.607000
1331.728000
1331.848000
1331.969000
1332.089000
```

The spectral plot of the data obtained from interferogram 20514D.ASC after using the software routines just described is shown in Figure 13.

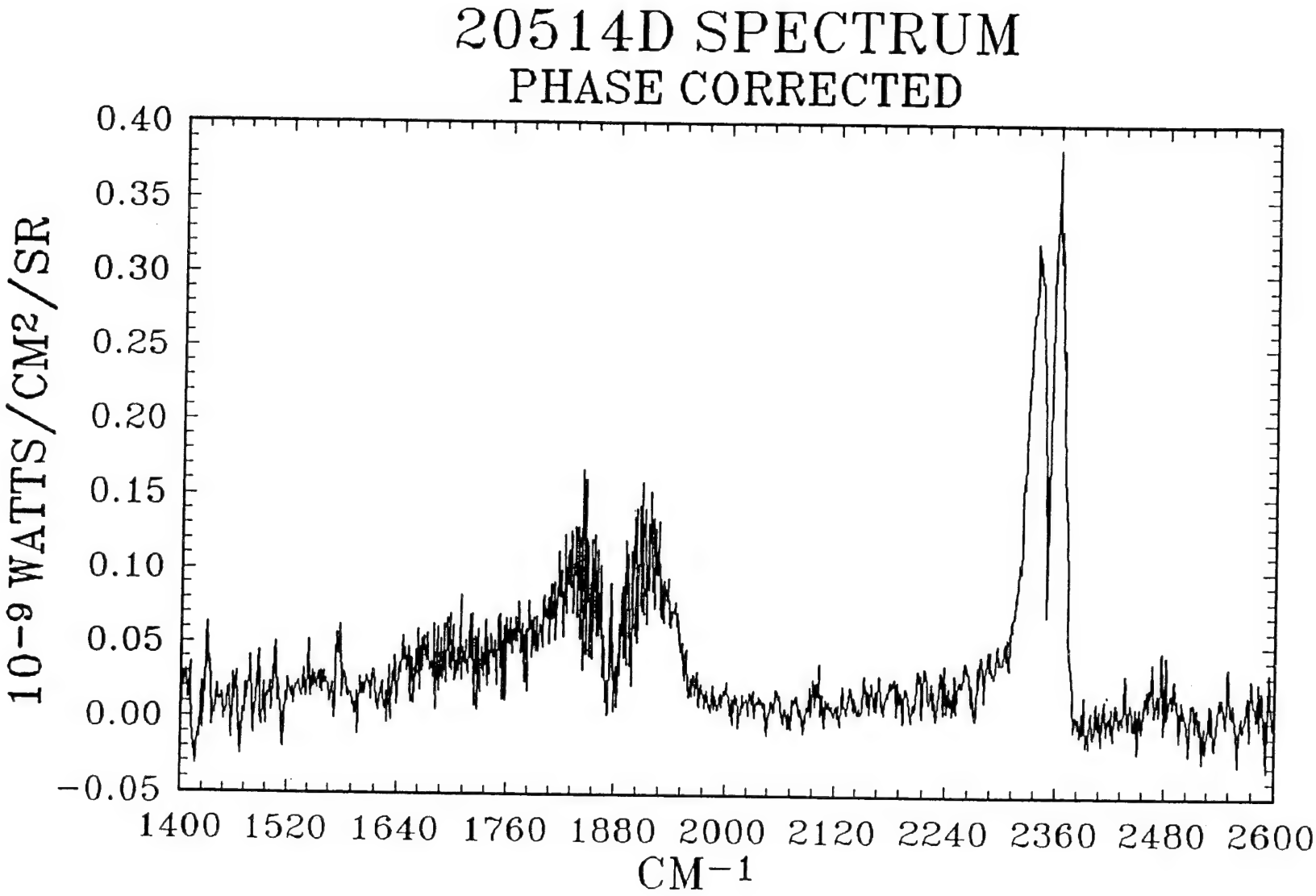


Figure 13. Recovered Spectrum of Electron Excited Air from the EXCEDE III Experiment.

9. SUMMARY AND CONCLUSIONS

The EXCEDE III rocket borne experiment succeeded in collecting a great deal of data on the response of the atmosphere to intense energetic electron bombardment. Interferometer data of the resulting emissions was collected over the altitudes of 80 to 115 km. Several other instruments on-board the sensor module collected additional data essential to the analysis of the emissions. The interferometer, its operation and the nature of the raw data, recorded as interferograms, was described briefly. The digital data (interferograms) require considerable processing in order to generate the final spectral product. The functions which must be performed on an interferogram were discussed. The goal of the effort reported here was to provide a simple desk top capability using a personal computer to perform all the necessary functions desired of a scientist wishing to investigate the EXCEDE III interferometer spectra.

A compact software package, delivered to PL/GPO has been developed which will perform the following; 1) phase correct interferograms, 2) co-add any number of interferograms, 3) Fourier transform a selected interferogram, 4) allow for Rectangular, Triangular or Kaiser Bessel (alpha 0-3) apodization, 5) interpolate output points in the recovered spectra using zero fill of factors of 2 and 4 in the interferogram which is then Fourier transformed, 5) permits pre-spectral filtering of the interferogram by setting filter band-pass edges, 6) provides spectral output files in different formats which are very useful for plotting programs, 7) is completely interactive with screen prompts. The software programs used in this package are provided in the appendices. An example of the capability showing the interaction between an operator and the computer is provided. A tutorial demonstrating the utility and ease of operation for this software has been given at GL/GPO.

REFRENCES

1. O'Neil, R.R., A. T. Stair, Jr., W.R. Pendelton, Jr., and D. Burt, The EXCEDE Spectral Artificial Auroral Experiment: An Overview, in Artificial Particle Beams, Space Plasma Studies, Ed. B. Grandal, 207-215, Plenum, New York, 1982.

2. Paulsen, D. E., P S. Armstrong, S. J. Lipson, J. Lowell, W. A. M. Blumberg, M. E. Fraser, W. T. Rawlins, B. D. Green, R. E. Murphy, R. Bien, J. W. Duff, R. J. Rieder, R. A. Armstrong, and M. H. Bruce, Overview of Infrared Observations from EXCEDE III, Proc. IRIS Targets, Backgrounds and Discrimination, 1, 91-103, 1994.

3. Thurgood, V. A., and R. J. Huppi, A Cryogenic Infrared Dual Optical Channel Interferometer-spectrometer for Upper Atmospheric Measurements, SPIE, 1145, 591-594, 1989.

4. Vanasse, G. A., and H. Sakai, Spectral Recovery in Fourier Spectroscopy, J. Opt. Soc. Amer., 58, 1, 84-92, 1968.

5. Foreman, M. L., W. H. Steel and G. A. Vanasse, Correction of Asymmeteric Interferograms Obtained in Fourier Spectroscopy, J. Opt. Soc. Amer. 56, 59-63, 1966.

6. Mertz, L., Transformations in Optics, John Wiley & Sons, Inc., New York, NY. 1967.

APPENDICES

APPENDIX A: Fortran Program REM

APPENDIX B: Fortran Program Co-add

APPENDIX A Fortran Program REM

```
      $large : s
           dimension d(16384),dd(16384),w(2048),filt(4,2),cc(2048)
           complex s(32768)
           integer iwk(15)
           character*64 infile,outfile,calfile,tempfile
           character qqq*1,dummy*1,ext*3
           logical exists
           pi=3.14159265
           nr=80
           icont1=0
           icont2=0
           icont3=0
           icont4=0
           icont5=0
           icont6=0
           ddx=-1.0
           fstart=0.0
           fend=8000.0
           inquire(file='debug.dat',exist=exists)
           if(.not.exists) goto 951
           open(unit=nr,file='debug.dat',status='old')
           rewind(nr)
           read(nr,*) icont1,icont2,icont3,icont4,icont5
          1,icont6
           if(icont6.gt.1) read(nr,*) ddx
       959 close(nr)
           if(icont6.gt.1) icont6=1
       951 continue
      c***** start *****
           inquire(file='cal1.asc',exist=exists)
           if(.not.exists) then
            write(*,*)
            write(*,*) '***** Calibration file CAL1.ASC missing *****'
            goto 950
           endif
           inquire(file='cal2.asc',exist=exists)
           if(.not.exists) then
            write(*,*)
            write(*,*) '***** Calibration file CAL2.ASC missing *****'
            goto 950
           endif
           inquire(file='cal3.asc',exist=exists)
           if(.not.exists) then
            write(*,*)
            write(*,*) '***** Calibration file CAL3.ASC missing *****'
            goto 950
           endif
```

```
      inquire(file='cal4.asc',exist=exists)
      if(.not.exists) then
       write(*,*)
       write(*,*) '***** Calibration file CAL4.ASC missing *****'
       goto 950
      endif
      inquire(file='filter.dat',exist=exists)
      if(.not.exists) then
       write(*,*)
       write(*,*) '***** Filter band file FILTER.DAT missing *****'
       goto 950
      endif
      open(unit=nr,file='filter.dat',status='old')
      rewind(nr)
      do 955 i=1,4
       read(nr,*,err=956,end=956) (filt(i,j),j=1,2)
  955 continue
      goto 900
  956 write(*,*)
      write(*,*) '***** Error reading filter band file FILTER.DAT *****'
      goto 950
c***** start of loop *****
  900 continue
      write(*,*)
      write(*,*) 'Input interferogram file name to read'
      read(*,'(a64)') infile
      inquire(file=infile,exist=exists)
      if(.not.exists) then
       write(*,*)
       write(*,*) '***** File does not exist *****'
       goto 900
      endif
      outfile=infile
      write(*,*)
  990 write(*,*) 'Input channel # (1-4)'
      read(*,*) ichan
      if(ichan.gt.4) goto 990
      if(ichan.le.0) goto 990
      if(ichan.eq.1) then
       fstart=filt(1,1)
       fend=filt(1,2)
       calfile='cal1.asc'
      elseif(ichan.eq.2) then
       fstart=filt(2,1)
       fend=filt(2,2)
       calfile='cal2.asc'
      elseif(ichan.eq.3) then
       fstart=filt(3,1)
```

```
       fend=filt(3,2)
       calfile='cal3.asc'
      elseif(ichan.eq.4) then
       fstart=filt(4,1)
       fend=filt(4,2)
       calfile='cal4.asc'
      endif
      ilen=len(infile)
      do 910 i=ilen,1,-1
       if(infile(i:i).eq.'.') goto 915
  910 continue
  915 ilen=i
      iread=0
      if(infile(ilen+1:ilen+1).eq.'C') infile(ilen+1:ilen+1)='c'
      if(infile(ilen+2:ilen+2).eq.'O') infile(ilen+2:ilen+2)='o'
      if(infile(ilen+3:ilen+3).eq.'R') infile(ilen+3:ilen+3)='r'
      if(infile(ilen+1:ilen+1).eq.'I') infile(ilen+1:ilen+1)='i'
      if(infile(ilen+2:ilen+2).eq.'N') infile(ilen+2:ilen+2)='n'
      if(infile(ilen+3:ilen+3).eq.'G') infile(ilen+3:ilen+3)='g'
      ext=infile(ilen+1:ilen+3)
      if(ext.eq.'cor'.or.ext.eq.'ing') iread=1
      write(*,*)
      write(*,*) 'reading ',infile
      open(unit=nr,file=infile,status='old')
      rewind(nr)
      if(iread.eq.1) goto 127
      do 125 i=1,10
       read(nr,'(a1)') dummy
  125 continue
  127 i=1
   10 if(iread.eq.0) read(nr,*,err=10,end=15) d(i)
      if(iread.eq.1) read(nr,*,err=10,end=15) xi,d(i)
      i=i+1
      if(i.gt.16384) goto 15
      goto 10
   15 close(nr)
      itotal=i-1
      write(*,*) itotal,' points read'
c     dx=6.328e-5
      dx=1.0/15798.017
      if(ichan.ge.3) dx=dx*2
      if(ddx.gt.0.0) dx=ddx
      nw=90
c***** write interferogram *****
      outfile(ilen+1:ilen+3)='ing'
      open(unit=nw,file=outfile)
      rewind(nw)
      do 815 i=1,itotal
```

```
      write(nw,*) i,d(i)
  815 continue
      close(nw)
      write(*,*)
      write(*,*) 'Interferogram file is ',outfile
  920 continue
      write(*,*)
      write(*,*) '    Input selection by number'
      write(*,*)
      write(*,*) '*********************************'
      write(*,*) '* 1 : phase correct             *'
      write(*,*) '* 2 : get uncalibrated spectrum  *'
      write(*,*) '* 3 : get calibrated spectrum    *'
      write(*,*) '* 4 : get new interferogram      *'
      write(*,*) '* 5 : quit                       *'
      write(*,*) '*********************************'
      write(*,*)
      read(*,*) isel
      if(icont6.eq.1.and.isel.eq.0) goto 930
      if(isel.le.0.or.isel.gt.5) then
       write(*,*) '***** Wrong selection, please select again *****'
       goto 920
      endif
      if(isel.eq.5) goto 950
      if(isel.eq.4) goto 900
      if(isel.ge.2) goto 940
  930 continue
      do 870 i=1,itotal
       dd(i)=d(i)
  870 continue
c***** get center portion *****
c      write(*,*) 'getting center piece'
      dp=0.0
      do 830 i=1,itotal
       if(dd(i).gt.dp) then
        dp=dd(i)
        icen=i
       endif
  830 continue
c***** limit section to < 256 points *****
      if(icen.gt.128) then
       ishift=icen-128
       itot2=itotal-ishift
       do 840 i=1,itot2
        dd(i)=dd(i+ishift)
  840  continue
       icen=icen-ishift
      endif
```

```
      ileft=icen-1
      istart=icen-ileft
      iend=icen+ileft
c      write(*,*) istart,icen,iend,dp
      do 30 i=istart,iend
       dd(i+1-istart)=dd(i)
   30 continue
      itot=iend-istart+1
      m=5
   26 m=m+1
      nn=2**m
      n=nn/2
      if(nn.lt.itot) goto 26
c      write(*,*) 'array size is ',nn
      do 35 i=1,nn
       s(i)=0.0
   35 continue
      df=1.0/nn/dx
      icen=ileft+1
      if(isel.eq.0) goto 935
      do 40 i=1,icen
       dd(i+icen)=dd(i+icen)*float(icen-i)/float(icen)
       dd(i)=dd(i)*float(i)/float(icen)
   40 continue
      if(icont1.ne.1) goto 952
      outfile='ingram1.dat'
      open(unit=nw,file=outfile)
      rewind(nw)
      do 45 i=1,itot
       write(nw,*) i+istart-1,dd(i)
   45 continue
      close(nw)
  952 continue
      s(1)=dd(icen)
      do 50 i=1,icen-1
       s(i+1)=dd(i+icen)
       s(nn+i-icen+1)=dd(i)
   50 continue
c      write(*,*) 'getting low res. phase, spectrum.'
      call realfft(s,m,iwk)
      if(icont2.ne.1) goto 953
      outfile='spect.dat'
      open(unit=nw,file=outfile)
      rewind(nw)
      do 60 i=1,n
       f=(i-1.0)*df
       if(f.lt.fstart.or.f.gt.fend) goto 60
       write(nw,*) f,cabs(s(i))
```

```
 60 continue
    close(nw)
953 continue
    if(icont3.eq.1) then
     outfile='phase.dat'
     open(unit=nw,file=outfile)
     rewind(nw)
    endif
    con=180.0/pi
    do 70 i=1,n
     f=(i-1.0)*df
      dd(i)=0.0
     if(f.lt.fstart.or.f.gt.fend) goto 70
     if(real(s(i)).eq.0.0) then
      dd(i)=pi/2.0
     else
      dd(i)=atan2(aimag(s(i)),real(s(i)))
     endif
     if(icont3.eq.1) write(nw,*) f,dd(i)*con
 70 continue
    if(icont3.eq.1) close(nw)
935 continue
c     write(*,*) 'getting phase interferogram'
    s(1)=0.0
    z=-1.0
    do 80 i=2,n+1
     s(i)=0.0
     s(nn+2-i)=0.0
     f=(i-1.0)*df
     if(f.lt.fstart.or.f.gt.fend) goto 80
     if(isel.eq.0) then
      s(i)=1.0
      s(nn+2-i)=1.0
     else
      s(i)=cmplx(cos(dd(i)),z*sin(dd(i)))
      s(nn+2-i)=conjg(s(i))
     endif
 80 continue
    call hermfft(s,m,iwk)
    do 85 i=1,nn
     dd(i)=0.0
 85 continue
    do 90 i=1,icen
     dd(icen+i-1)=real(s(i))/nn
     if(i.eq.icen) goto 90
     dd(i)=real(s(nn+1-icen+i))/nn
 90 continue
    alpha=3.0
```

```
      beta=pi*alpha
      scale=beso(beta)
      idelta=nn-icen-1
      delta=float(idelta)
      ipeak=icen
      do 94 i=1,idelta
       temp=(i-1.0)/delta
       temp=sqrt(1.0-temp*temp)
       apod=beso(beta*temp)/scale
       dd(i+ipeak-1)=dd(i+ipeak-1)*apod
       if(i.eq.1.or.i.gt.ipeak) goto 94
       dd(ipeak+1-i)=dd(ipeak+1-i)*apod
   94 continue
      if(icont4.ne.1) goto 954
      outfile='corr.dat'
      open(unit=nw,file=outfile)
      rewind(nw)
      do 95 i=1,itot
       write(nw,*) i,dd(i)
   95 continue
      close(nw)
  954 continue
c      write(*,*) 'phase correcting'
      outfile=infile
      outfile(ilen+1:ilen+3)='cor'
      open(unit=nw,file=outfile)
      rewind(nw)
      do 110 i=1,itotal-itot
       sum=0.0
       do 120 j=1,itot
        ij=i+j-1
        if(ij.le.0) goto 120
        if(ij.gt.itotal) goto 115
        sum=sum+d(ij)*dd(j)
   120  continue
   115  write(nw,*) i,sum
   110 continue
      close(nw)
      write(*,*)
      write(*,*) 'Phase corrected interferogram is ',outfile
      itotal=itotal-itot
      infile=outfile
      open(unit=nr,file=infile,status='old')
      rewind(nr)
      do 130 i=1,itotal
       read(nr,*,end=135) j,d(i)
   130 continue
   135 itotal=j
```

```
      close(nr)
      goto 920
  940 continue
      write(*,*)
      write(*,*) ' Choose apodization by number'
      write(*,*)
      write(*,*) '++++++++++++++++++++++++++++++++'
      write(*,*) '+    0 : Rectangular       +'
      write(*,*) '+    1 : Triangular        +'
      write(*,*) '+    2 : Kaiser Bessel     +'
      write(*,*) '++++++++++++++++++++++++++++++++'
      write(*,*)
      read(*,*) iapod
      if(iapod.lt.0.or.iapod.gt.2) then
       write(*,*) 'Wrong selection'
       goto 940
      endif
      dp=0.0
      do 150 i=1,itotal
       if(d(i).gt.dp) then
        dp=d(i)
        ipeak=i
       endif
  150 continue
c      write(*,*) ipeak,dp
      do 155 i=1,itotal
       dd(i)=d(i)
  155 continue
c-----------
c      alpha=0.0
c      call ksramp(ipeak,dd,itotal,alpha)
c      call ksright(ipeak,dd,itotal,alpha)
c----------
      delta=itotal-ipeak
      idelta=itotal-ipeak
      if(iapod.eq.1) then
       do 210 i=1,idelta
        dd(i+ipeak-1)=dd(i+ipeak-1)*(delta+1.0-i)/delta
        if(i.ge.ipeak) goto 210
        dd(ipeak-i)=dd(ipeak-i)*(delta-i)/delta
  210  continue
      elseif(iapod.eq.2) then
       write(*,*) 'Input alpha (0-3)'
       read(*,*) alpha
       beta=pi*alpha
       scale=beso(beta)
       do 220 i=1,idelta
        temp=(i-1.0)/delta
```

```
      temp=sqrt(1.0-temp*temp)
      apod=beso(beta*temp)/scale
      dd(i+ipeak-1)=dd(i+ipeak-1)*apod
      if(i.eq.1.or.i.gt.ipeak) goto 220
      dd(ipeak+1-i)=dd(ipeak+1-i)*apod
220   continue
    endif
222 write(*,*)
    write(*,*) 'Input factor to interpolate (1,2,4)'
    read(*,*) inter
    if(inter.lt.0) goto 222
    if(inter.eq.0) inter=1
    if(inter.eq.3) goto 222
    if(inter.gt.4) goto 222
    ilarge=0
    m=5
126 m=m+1
    nn=2**m
    n=nn/2
    if(m.gt.15) ilarge=ilarge+1
    if(n.lt.inter*idelta) goto 126
    df=1.0/nn/dx
128 continue
    jlarge=2**ilarge-1
    write(*,*) 'Array size ',nn
    if(m.gt.15) then
     m=15
     nn=2**m
     n=nn/2
     df=1.0/nn/dx
    endif
    do 160 i=1,nn
     s(i)=0.0
160 continue
    do 165 i=1,idelta
     s(i)=dd(i+ipeak-1)
     if(i.eq.1) goto 165
     s(nn+2-i)=s(i)
165 continue
    call realfft(s,m,iwk)
    tempfile='out0.dat'
    open(unit=nw,file=tempfile,form='unformatted')
    rewind(nw)
    do 820 i=1,n
     ans=real(s(i))
     f=(i-1.0)*df
     if(f.lt.fstart.or.f.gt.fend) goto 820
     write(nw) ans
```

```
 820 continue
     close(nw)
     if(ilarge.ge.1) then
      do 600 kkk=1,jlarge
       do 810 i=1,nn
        s(i)=0.0
 810   continue
       dpi=2.0*pi/float(nn)*float(kkk)/(jlarge+1.0)
       do 850 i=1,idelta
        ddpi=dpi*(i-1.0)
        s(i)=dd(i+ipeak-1)*cmplx(cos(ddpi),sin(ddpi))
        if(i.eq.1) goto 850
        s(nn+2-i)=conjg(s(i))
 850   continue
       call hermfft(s,m,iwk)
       write(qqq,'(i1)') kkk
       tempfile(4:4)=qqq
       open(unit=kkk,file=tempfile,form='unformatted')
       rewind(kkk)
       do 860 i=1,n
        ans=real(s(i))
        f=(i-1.0)*df
        if(f.lt.fstart.or.f.gt.fend) goto 860
        write(kkk) ans
 860   continue
       close(kkk)
 600   continue
     endif
     if(isel.eq.3) then
      nnr=50
      write(*,*) 'Reading cal. file  ',calfile
      open(unit=nnr,file=calfile,status='old')
      rewind(nnr)
      i=1
      read(nnr,*)
      read(nnr,*)
c      read(nnr,*)
c      read(nnr,*)
  180 read(nnr,*,end=185,err=180) w(i),cc(i)
      i=i+1
      goto 180
  185 jtot=i-1
      close(nnr)
c      write(*,*) jtot,w(1),cc(1),w(jtot),cc(jtot)
     endif
     outfile=infile
     if(isel.eq.2) outfile(ilen+1:ilen+3)='spu'
     if(isel.eq.3) outfile(ilen+1:ilen+3)='spc'
```

```
      tempfile='out0.dat'
      open(unit=nr,file=tempfile,form='unformatted',status='old')
      rewind(nr)
      if(ilarge.ge.1) then
       do 610 kkk=1,jlarge
        write(qqq,'(i1)') kkk
        tempfile(4:4)=qqq
        open(unit=kkk,file=tempfile,form='unformatted',status='old')
        rewind(kkk)
  610  continue
      endif
      open(unit=nw,file=outfile)
      rewind(nw)
      nlarge=n
      do 170 i=1,nlarge
       f=(i-1.0)*df
       cal=1.0
       if(f.lt.fstart.or.f.gt.fend) goto 170
       if(isel.eq.3) then
        if(f.le.w(1).or.f.ge.w(jtot)) goto 197
        do 190 j=1,jtot
         if(w(j).gt.f) goto 195
  190   continue
  197   cal=0.0
        goto 196
  195   cal=(f-w(j-1))*(cc(j)-cc(j-1))/(w(j)-w(j-1))
        cal=cal+cc(j-1)
        cal=1.0/cal
  196   continue
       endif
       read(nr) ans
       write(nw,*) f,ans*dx*cal
       if(ilarge.ge.1) then
        do 620 kkk=1,jlarge
         read(kkk) ans
         fff=f+df*float(kkk)/(jlarge+1.0)
         if(isel.eq.3) then
          if(f.le.w(1).or.f.ge.w(jtot)) then
           cal=0.0
          else
           cal=(fff-w(j-1))*(cc(j)-cc(j-1))/(w(j)-w(j-1))
           cal=cal+cc(j-1)
           cal=1.0/cal
          endif
         endif
         write(nw,*) fff,ans*dx*cal
  620   continue
       endif
```

```
  170 continue
      do 630 kkk=1,jlarge
       close(kkk)
  630 continue
      close(nr)
      close(nw)
      write(*,*)
      write(*,*) 'Spectrum file is ',outfile
      goto 920
  950 continue
      stop
      end

      subroutine ksramp(icent,b,len,alpha)
c***does 'left-ramp' portion of a Kaiser apodization
c***affects b(1) thru b(2*icen-1)
      real b(1)
      b(1)=b(1)*0.5
      if(icent.eq.1) return
      pi=3.14159265
      beta=pi*alpha
      relen=1.0/(len-icent)
      anorm=beso(beta)
      max=2*icent-1
      eta=2*(icent-1)
      temp=sqrt(1.0-(eta*relen/2.0)**2)
      temp=beso(beta*temp)/anorm
      slope=temp/eta
      do 10 k=1,max
       diff=k-icent
       temp=diff*relen
       temp=sqrt(1.0000001-temp*temp)
       apo=diff*slope+0.5*beso(beta*temp)/anorm
       b(k)=b(k)*apo
   10 continue
      return
      end
c
      subroutine ksright(icent,b,len,alpha)
c***perform right-sided portion of Kaiser apodization
c***affects b(2*icen) thru b(len)
      real b(1)
      pi=3.14159265
      beta=pi*alpha
      relen=1.0/(len-icent)
      anorm=beso(beta)
      do 10 k=2*icent,len
       diff=k-icent
```

```
      temp=diff*relen
      temp=sqrt(1.0000001-temp*temp)
      apo=beso(beta*temp)/anorm
      b(k)=b(k)*apo
   10 continue
      return
      end
c
      function beso(x)
      parameter(t=1.0e-8)
      y=x/2.0
      e=1.0
      de=1.0
      do 1 i=1,25
       de=de*y/i
       sde=de*de
       e=e+sde
       if(e*t-sde) 1,1,2
    1 continue
    2 beso=e
      return
      end
```

APPENDIX B Fortran Program COADD

```
      dimension d1(16384),d2(16384),d3(16384)
      character*64 infile1,file2
      nr=10
      open(unit=nr,file='datfile.dat',status='old')
      rewind(nr)
      read(nr,*) itotal
      read(nr,'(a64)') infile1
      itotal=itotal-1
      write(*,*) 'Reading ',infile1
      call reader(infile1,d1,iii)
      dp=0.0
      ipeak1=1
      do 10 i=1,iii
       if(d1(i).gt.dp) then
        dp=d1(i)
        ipeak1=i
       endif
   10 continue
      ileft1=ipeak1-1
      iright1=iii-ipeak1
      write(*,*) ileft1,ipeak1,iright1
      write(*,*) iii,'  points read'
      do 20 i=1,itotal
       read(nr,'(a64)') file2
       write(*,*) 'reading ', file2
       call reader(file2,d2,jjj)
       write(*,*) jjj,'  points read'
       dp=0.0
       ipeak2=1
       do 30 j=1,jjj
        if(d2(j).gt.dp) then
         dp=d2(j)
         ipeak2=j
        endif
   30  continue
       ileft2=ipeak2-1
       iright2=jjj-ipeak2
       write(*,*) ileft2,ipeak2,iright2
       if(ileft2.lt.ileft1) ileft1=ileft2
       if(iright2.lt.iright1) iright1=iright2
       iii=iright1+1+ileft1
       do 40 k=1,iii
        d3(k)=d1(ipeak1-ileft1+k-1)
   40  continue
       dp=0.0
       do 50 k=1,iii
        d1(k)=d3(k)+d2(ipeak2-ileft1+k-1)
```

```
      if(d1(k).gt.dp) then
       dp=d1(k)
       ipeak=k
      endif
 50  continue
     ipeak1=ileft1+1
     write(*,*) ileft1,ipeak1,iright1,ipeak
 20 continue
    close(nr)
    nw=50
    write(*,*) 'Writing coadded file out1.dat'
    open(unit=nw,file='out1.dat')
    rewind(nw)
    do 60 i=1,iii
     write(nw,*) d1(i)/(itotal+1.0)
 60 continue
    close(nw)
    write(*,*) 'Writing coadded file out2.ing'
    open(unit=nw,file='out2.ing')
    rewind(nw)
    do 70 i=1,iii
     write(nw,*) i,d1(i)/(itotal+1.0)
 70 continue
    close(nw)
    stop
    end
    subroutine reader(infile1,d1,iii)
    dimension d1(1)
    character*64 infile1
    ilen=len(infile1)
    do 10 i=ilen,1,-1
     if(infile1(i:i).eq.'.') goto 15
 10 continue
 15 inum=0
    ilen=i
    if(infile(ilen+1:ilen+1).eq.'C') infile(ilen+1:ilen+1)='c'
    if(infile(ilen+2:ilen+2).eq.'O') infile(ilen+2:ilen+2)='o'
    if(infile(ilen+3:ilen+3).eq.'R') infile(ilen+3:ilen+3)='r'
    if(infile(ilen+1:ilen+1).eq.'I') infile(ilen+1:ilen+1)='i'
    if(infile(ilen+2:ilen+2).eq.'N') infile(ilen+2:ilen+2)='n'
    if(infile(ilen+3:ilen+3).eq.'G') infile(ilen+3:ilen+3)='g'
    if(infile(i+1:i+3).eq.'ing'.or.infile(i+1:i+3).eq.
   1'cor') inum=1
    nrr=20
    open(unit=nrr,file=infile1,status='old')
    rewind(nrr)
    if(inum.eq.0) then
     do 30 i=1,10
```

```
    read(nrr,*)
30  continue
   endif
   i=1
20 if(inum.eq.0) then
    read(nrr,*,err=20,end=25) d1(i)
   else
    read(nrr,*,err=20,end=25) xj,d1(i)
   endif
   i=i+1
   if(i.gt.16384) goto 25
   goto 20
25 close(nrr)
   iii=i-1
   return
   end
```